



UiT The Arctic University of Norway

Faculty of Science and Technology
Department of Computer Science

Nudging according to user's preferences

Sufyan Saleem

INF-3990 Master's Thesis in Computer Science – November 2022



Table of Contents

- 1 Introduction 11
 - 1.1 Background..... 11
 - 1.2 Problem Statement..... 13
 - 1.3 Purpose 13
 - 1.4 Goal 13
 - 1.5 Related Work..... 14
 - 1.6 Physical Inactivity 14
 - 1.7 User preferences 16
 - 1.8 Methodology..... 16
 - 1.9 Delimitations 16
 - 1.10 Outline..... 17
- 2 Nudge 19
 - 2.1 Digital Nudging 19
 - 2.2 Smart Nudging..... 19
 - 2.3 Designing a Digital Nudge 20
 - 2.3.1 Define the goal 20
 - 2.3.2 Understand the user..... 22
 - 2.3.3 Design the nudge 22
 - 2.3.4 Test the Nudge 23
 - 2.4 Nudging user..... 23
- 3 Recommendation Systems 27
 - 3.1 Types of Recommendation Systems..... 27
 - 3.1.1 Content-Based Filtering 28
 - 3.1.2 Collaborative Filtering 28

3.1.3	Collaborative Filtering Challenges.....	28
3.1.4	Hybrid System.....	30
3.1.5	Our objective and Recommender systems	30
3.2	Collaborative Filtering sub-categories.....	30
3.2.1	Memory-Based Collaborative Filtering	31
3.2.2	Model-Based Collaborative Filtering.....	31
3.2.3	User-Based collaborative filtering	32
3.2.4	Item-Based collaborative filtering.....	34
4	Tools and Technology	37
4.1	Fitbit.....	37
4.1.1	Fitbit API's.....	37
4.2	Firebase.....	38
4.3	React Native	40
4.3.1	Expo	40
4.3.2	Expo Go.....	41
4.4	Python.....	41
5	Methods and Methodologies	43
5.1	Research Methods and Methodologies.....	43
5.2	Data collection and Analysis	44
5.2.1	Dataset/Data collection	44
5.2.2	Data Analysis	45
5.3	Quality Assurance.....	46
5.4	Software development method	47
5.5	Evaluation of the Recommendation model	47
6	Requirements and Application Design.....	49

6.1	Requirements	49
6.2	Application Design	49
6.3	Mobile Application.....	50
6.3.1	Cross-platform.....	50
6.3.2	Connection to Fitbit.....	50
6.3.3	Sending Notification to Fitbit watch	50
6.3.4	Rating activities.....	51
6.3.5	Recommendation.....	51
6.3.6	User Interface	51
6.3.7	Intended and Fitbit App.....	57
6.3.8	Sources of Data	58
7	Implementation of the System.....	59
7.1	Implementation of User-Based Recommendation System.....	59
7.2	Implementation of Item Based Recommendation System	60
7.3	Evaluation of the Model	60
7.4	Recommendation System API.....	61
7.5	Nudge Script.....	62
7.6	Nudging API.....	63
7.7	Firestore	63
7.8	Cloud Function	65
7.8.1	Expo Push Notification	66
7.9	Flow of the whole system.....	68
8	System Testing	71
8.1	Results	71
9	Future Work and Conclusion	73

9.1	Overview Screen.....	73
9.2	Historical Data Screen.....	74
9.3	Conclusion.....	75
	Works cited.....	77

List of Tables

Table 1	Potential steps/day category wise.....	15
Table 2	Selection of nudge principles, descriptions, and examples.....	21

List of Figures

- Figure 1 Life Cycle of Digital Nudge 20
- Figure 2 Nudge Attributes..... 24
- Figure 3 Pearson correlation 33
- Figure 4 Rating prediction (user-based)..... 33
- Figure 5 Cosine Distance 35
- Figure 6 Rating prediction (item-based) 36
- Figure 7 Activities Questionnaire 45
- Figure 8 Login Screen 52
- Figure 9 Registration Screen 53
- Figure 10 Select and Rate Activities Screen 53
- Figure 11 Set-up Daily Goal and Active hours 54
- Figure 12 Home Screen..... 55
- Figure 13 Profile Screen..... 56
- Figure 14 User profile setting screen 56
- Figure 15 Editing Screen..... 57
- Figure 16 User-Based Implementation 59
- Figure 17 Email Format 61
- Figure 18 Expo Push Notification..... 67
- Figure 19 Overview Screen..... 73
- Figure 20 Historical Data Screen 74

Abstract

Physical inactivity has been identified as a global pandemic, physical inactivity causes multiple health outcomes in different demographic group such as coronary heart disease, type 2 diabetes, colon, and breast cancer. A physical inactive person takes less than 5000 steps a day. To try to reduce physical inactivity along individual for healthy lifestyle, this thesis provides personalized digital nudge. Nudge means to guide someone to do something that is beneficial for the long-term benefit of the person being nudged and doing so using UI (user interface) in digital environment is known as digital nudging. As people are relying more on technology for their decision making, the information collected from the integration of the devices is used to provide personalized nudges. As people have access to smartphones and wearable devices, data is collected from these devices to provide tailored nudges to achieve minimum required steps to reduce inactivity. Personalized nudge is a smart nudge which predictably influence people's behaviour. It is a type of digital nudge. This kind of nudge takes user's information into account before nudging a user. This thesis also provides recommendations (new activities) based on person's preference. The presented system was also tested by real users, and the feedback suggested that the presented system indeed urged them to be more active.

Keywords: nudging, smart nudging, behavioral change, recommender system, physical inactivity

1 Introduction

Nudging [1] means to motivate someone gently or to encourage or persuade someone to do something gently rather than forcefully. It enables people to make decisions that can help them make changes that can improve their lives [2].

A nudge is a tool [3] which can encourage people to live life according to their wishes but fails to realise due to automatic thinking. It is observed from a study [4] that people are influenced by the social and physical environment while making decisions. Therefore, the goal of a nudge is to change the behaviour of individuals in making decisions that are better for health, society, or environment-friendly behaviour through suggestions, reinforcement, and non-compulsion [5]

Physical inactivity [6] has been identified as a global pandemic and the fourth leading cause of death worldwide. Based on this evidence, inactivity has been the cause of many diseases such as coronary heart disease, type 2 diabetes, colon, and breast cancer, which result in 5.3 million deaths worldwide [6]. It is evident from research [7] that there should be a system that can help reduce the physical inactivity of people, and to achieve this, nudging seems promising and since it might make people exercise daily. Therefore, a multidisciplinary approach [8] is needed to combine both psychological perspectives along with current recommendation techniques to design such a system for behavioural change.

1.1 Background

The increased usage of smartphones by people has turned from necessity to compulsive usage leading to addiction [9]. To prevent this problematic use of smartphones the nudge theory in behavioural economics was applied to investigate if the usage of smartphones is reduced with nudges. A smartphone application was developed based on Thaler and Sunstein's nudge theory [10]. During the study [9] revealing the total screen time decreased when the users were nudged based on their smartphone usage. Although intermittent glancing at the smartphone, as well as the median session time of the smartphone, increased during the trial, as the score is earned during abstinence. A relapse in smartphone usage was observed after the discontinuation of

nudges. It is suggested that users respond to nudges differently. However, the nudges may help users with higher scores reduce their smartphone usage [9].

Excessive sedentary behaviour, that is activities that occur while sitting or lying down and require low energy expenditure [11], [12] has been associated with many adverse diseases like obesity [13], [14], poor cardiometabolic health [15], [16] and mortality [26], [27]. According to recent studies short intervals of light to moderate intensity walking can improve postprandial glucose and insulin responses [19], [20], energy expenditure [21], and gene expression in skeletal muscle that are involved in carbohydrate metabolism [22]. Although other intervention approaches [23],[24] have been shown to be more effective in reducing sedentary behaviour but they were limited to a workplace, expensive equipment or an intensive lifestyle. As most people have access to a smartphone, therefore present a great opportunity to monitor time spent sedentary as the smartphone has a built-in accelerometer. An approach of smartphone-based intervention was proposed to monitor sedentary behaviour in obese people [25].The approach was used in conjunction with SenseWear Mini Armband [26], [27] and prompts to physical activity after periods of sitting. A mobile application called B-Mobile [25] was used by presenting three physical activity break conditions: (1) 3-min break prompt after 30 minutes of sedentary behaviour; (2) 6-min break prompt after 60 minutes of sedentary behaviour; and (3) 12-min break prompt after 120 minutes of sedentary behaviour for 7 days. The results concluded that for overweight or obese people, frequent short activity breaks might be most effective for decreasing sedentary behavior. Further studies must be done to evaluate if the results can be maintained for the long term [25].

The use of gamification to promote physical activity and persuade people to live healthy and active lifestyles has emerged in the last few years and is termed an exergame [28]. Most persuasive games in health and wellness are based on the approach of "one-size-fits-all", which is shown to be suboptimal [29] Therefore, a combination of models based on personality types and adaptive recommendations with gamified elements is needed. A comprehensive model was proposed during the study [30] that uses detailed and dynamic player modelling and wearable-based tracking to provide personalised game features and activity recommendations.

The above studies showed that nudging is an effective way to change a person's behaviour; it can be used in any context with less smartphone usage and physical activity.

1.2 Problem Statement

As physical inactivity [6] has been identified as a global pandemic and the fourth leading cause of death worldwide, to tackle it, there is yet no system that can present physically inactive users with a personalised nudge, and that is why this thesis presents a system that is intended to provide the personalised nudge that takes into account user's preference such as (user's active hours, preferred activity) and nudge a user that is physically inactive for a prolonged time. The system will also recommend new physical activities based on a user's already rated activities so a user can try new activities that they might like to try.

1.3 Purpose

The purpose of this thesis is to explain and assess an idea whether or not a mobile application can be used to help reduce physical inactivity in individuals. This thesis describes the design and implementation of the system and is tested by real users to answer whether the system helped them in becoming more active.

1.4 Goal

The goal of this thesis is to develop a mobile application, which aim is to try to eliminate physical inactivity that has become the cause of many serious diseases as explained in section 1.3. It evaluates whether the mobile application can help in reducing physical inactivity. This thesis focuses on making tailored and informed nudges, where a user will be nudged with their preferred activity from a set of activities that the user will have pre-selected. This nudge will also take into account different attributes such as (user goal, user active timing of the day, and activities steps.). These nudges will not be enforcing obligations. Instead, they will be a gentle reminder to do the suggested activity or walk the remaining number of minimum steps. This thesis will also employ recommendations, to recommend new activities to user that they might like to try. After the implementation of the system, it will be tested by real users, and the results obtained will be presented in more detail. The results will be a bridge for the research community whether a sophisticated mobile application can help in reducing a serious problem which in this case is physical inactivity and whether human decisions can be driven by more personalized nudges.

1.5 Related Work

Some of the results from previous studies have shown that physical inactivity causes multiple health outcomes in different demographic groups [31]. Physical inactivity among the leading causes of death worldwide ranks as the fourth [6]. However, it is hard to find research reports that have investigated the use of personalised nudges to tackle physical inactivity.

Two persons may not react in a specific way to a nudge depending on various factors like time, location, situation, capability, personality, and preferences concerning intervention. It is a challenge to present a personalised nudge as it depends on many factors such as user environment, age, preferences etc. To facilitate behaviour change, nudges must be personalised to specific user preferences [5].

Previous studies [8] also indicate that the reason traditional recommender systems fail is that users change preferences and discontinue a type of activity before a new habit is adapted, so for intervention, personalisation is required.

1.6 Physical Inactivity

Physical inactivity [32] has been defined as "*not meeting a 30 min moderate activity for at least 5 days a week, 20 minutes of at least intense activity for 3 times a week*". It is stated that 100 steps/min for 30 minutes are recommended [33], though it is not enough to make it a threshold/measure which can be used as a base to initiate a nudge. To tackle this, the steps count per day seems to be more appropriate than working for 30 mins a day or doing 100 steps/min for 30 minutes. It is argued that 8000 steps/day is a suitable guideline for achieving public health physical activity recommendations [34]. Though, another study suggests that 10,000 steps/day is a good measure for staying physically active [35]. Apichai and Sanhapan [35] says in their paper that taking 10,000 steps a day more than three times a week is comparable to doing physical activity at moderate intensity for 30 minutes on most days of the week. By this accumulated evidence, it is safe to say that physically active phase can also be achieved with step counts. Though according to a study [33], it is recommended that adults take between 4,000 and 18,000 steps each day, and 10,000 steps is a reasonable goal for them to achieve. Figure 1 shows the minimum steps needed to be active, and we will use this figure that the study suggested [36] as a threshold for being physically active. In this thesis, the healthy adults' age

ranges from 18-64 years, according to the CDC [37] adults age ranges from 18 to 64 years of age.

Healthy Adults

<i>Steps/Day Zone</i>	<i>Descriptive Category</i>
$\geq 12,500$	Highly active
10,000 – 12,499	Active
7500 - 9999	Somewhat active
5000 - 7499	Low active
< 5000	Sedentary

Table 1 Potential steps/day category wise

To be considered active, a healthy adult need >5000 [36] steps per day, Table 1 demonstrates different categories based on steps taken.

- **Sedentary:** steps less than 5000 per day lie in this category.
- **Low active:** steps between 5000-7499 per day lies in this category.
- **Somewhat active:** steps between 7500-9999 per day lie in this category.
- **Active:** steps between 10,000-12,499 per day lie in this category.
- **Highly active:** steps more than 12,500 per day lie in this category.

As the goal of the thesis is to make people active, which means a person who takes more than 5,000 steps a day is active, though based on all other evidence (as explained above), the more steps the better. The user will also be nudged to achieve at least **Low active** state (this state is a default state, the user can select a goal greater than 5000 steps and user will be nudged till the user have reached their goal), with an option to stop getting nudge for the day after achieving minimum steps that are 5,000 steps per day.

1.7 User preferences

User Preferences [38] are the opinions of the user about goods, services, friends, advertisements, and search engine results related to social topics. User preferences are usually linked to recommender systems since a recommender system's goal is to convert data about users and their preferences into forecasts of their potential interests. In this thesis, recommended activity will be based on user preference suggested by a recommender model. The user will be presented with predefined activities and based on user selection and new similar activities will be recommended to the user. Therefore, when making recommendations for a specific user, the most basic strategy is to choose objects liked by other users that are similar to the target user [39].

1.8 Methodology

Research Methodology that can be used for the thesis is of two main types, quantitative and qualitative. The quantitative method verifies or falsifies theories and hypotheses, or the functionality and interfaces of computer systems through experiments and testing [40], while the qualitative method is, developing computer systems, artifacts, and inventions by understanding meanings, opinions, and behaviors [40]. In this thesis, qualitative research methodologies have been used to deploy the idea of personalised nudging to reduce physical inactivity and test whether it is effective in reducing physical inactivity in an individual. For evaluation of the system, user feedback was required to truly measure the effectiveness of the system.

1.9 Delimitations

This thesis will provide in-depth knowledge along with the practical implementation of the system that will be used to tackle inactivity in individuals. The overall aim was supposed to tackle the inactivity problem that has engulfed the 21st century population. The initial idea was to make a system that can nudge a user on a daily basis to do a suggested activity based on the user context along with environmental context such as user age, user height, user weight, weather, time of day. Unfortunately, it was deemed too broad for degree thesis.

Instead of context-based nudges, this thesis will provide personalised nudges of user-preferred activities based on user preference, and new activities will also be recommended to users using collaborative filtering techniques.

1.10 Outline

The following thesis topics are as follows: Section 2 focuses on Nudging, its types and designing. Section 3 presents the recommender model, types of recommender systems and process of selected recommender system. Section 4 contains different tools and technology used to develop the intended system. Section 5 discusses the methodology and methods employed during conducting this thesis. Section 6 points out the requirements and design of the Application, its working and user interface. Section 7 focuses on the implementation of the model, API's and mobile application communication. Section 8 contains the system evaluation and test result. Section 9 provides future work and discusses conclusion.

2 Nudge

Nudge is defined as “*any aspect of the choice architecture that alters people’s behaviour in a predictable way without forbidding any options or significantly changing their economic incentives*” [41] choice architecture refers to “*the environment in which individuals make choices*” [41]. The paper by Sandor and Randi [41] also states that, for an intervention to be considered a nudge, it must be simple and inexpensive. Nudges are not commands to a user nor a system, for example a nudge is when a user place fruit at eye level, instead of prohibiting certain foods not to eat. A nudge is any little intervention in complex decision-making settings that overcomes cognitive mistakes and guides people toward particular, advantageous options while leaving them free to make other choices [42]. A nudge, in other words, refers to subtle encouragement rather than direct enforcement. Physical inactive people will be nudged daily according to a predefined set of rules so that they can be physically active. The people nudged will have a choice to proceed or not proceed with the suggested activity.

2.1 Digital Nudging

In a common definition User interface (UI) [43] features of software applications that influence user choices which are referred to as digital nudging. Again, the general purpose is to assist users in making better selections for themselves, i.e., to direct them to a specified, preferred option without constraining their options. Digital nudging [5] is described as a “*subtle form of using design, information, and interactive elements to guide user behaviour in digital environments, without restricting the individual's freedom of choice*”. This thesis also focuses on digital nudging, since the UI elements of Android operating system will be used to influence the choices of the users. Though this thesis employs a more sophisticated digital nudge that is also known as Smart nudging.

2.2 Smart Nudging

Smart nudging employs cognitive technologies to predictably influence people's behaviour without restricting their choices or changing their economic motives [42]. The paper by Randi and Anders [5] defined smart nudging as follows: “*digital nudging, where the guidance of user behaviour is tailored to be relevant to the current situation of each individual user.*” Randi and Sandor refer to a personalized nudge as a smart nudge [41]. Personalisation and context

awareness are used to tailor smart nudges, with knowledge about the user and their situation being central. These nudges will not be enforcing obligations; instead, they will be a gentle reminder to do the suggested activity or walk the remaining number of minimum steps. These minimum steps will be based on Table 1.

2.3 Designing a Digital Nudge

Designing a digital nudge to nudge users involves a cycle that involves different steps to carefully create an effective nudge [44]. Figure 1 shows the life cycle of a digital nudge.

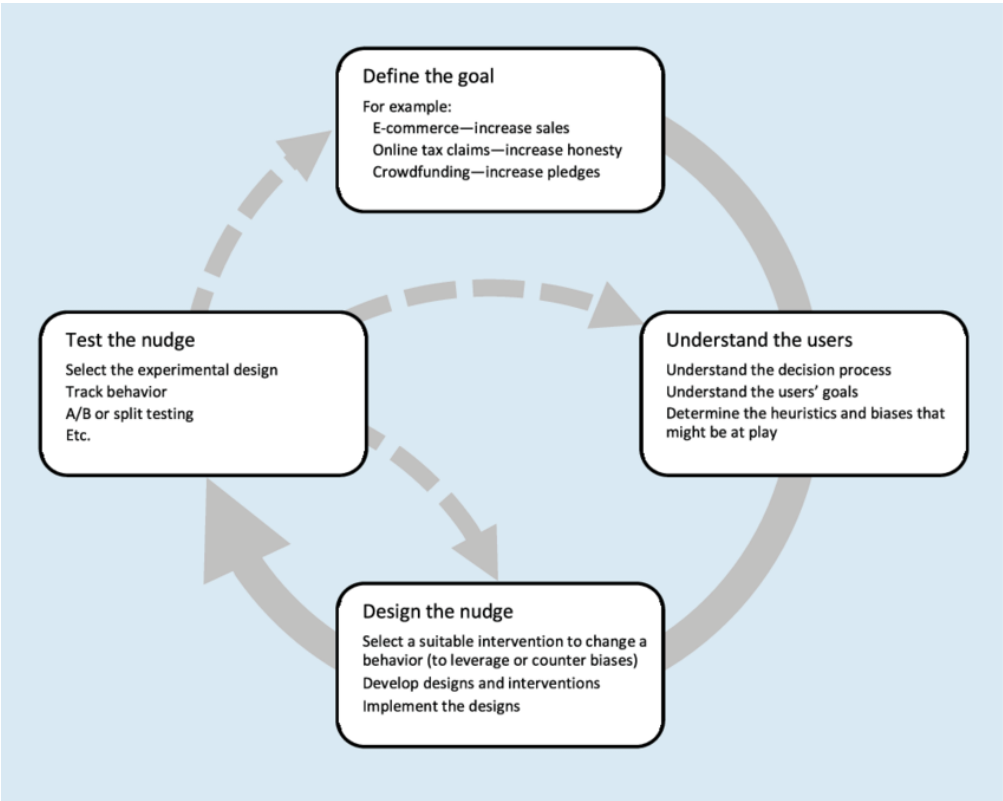


Figure 1 Life Cycle of Digital Nudge

According to Figure 1 [44], there are 4 steps involved in designing an effective nudge. This thesis will go through each step in the cycle in detail.

2.3.1 Define the goal

Design of the nudge and definition of the goal is involved in this step. The design of the system needs to employ the components that are needed to do a successful nudge; Table 2 illustrates the principles of nudge [45].

Nudge principles, descriptions, and examples		
Nudge Principles	Description	Example
Incentive	Enhancing the effectiveness of incentives by making them more salient	Telephones that are programmed to display the running costs of phone calls
Understanding mapping	The mapping of difficult-to-evaluate information to familiar evaluation schemes	In advertising a digital camera, megapixels are mapped to maximum printable size rather than being pointed to them
Defaults	Pre-selecting options by setting default option	Automatic renewal of subscriptions
Giving feedback	Giving feedback to users when they do well or make mistakes	Electronic road signs with smiling or sad faces depending on the vehicle's speed
Expecting error	Expecting users to make errors and being as forgiving as possible	Requiring people at an ATM to retrieve the card before they receive their money in order to help them avoid forgetting the card
Structure complex choices	All the alternatives should be listed, and people should be allowed to make trade-offs when necessary	Online product configuration systems that make choices simpler by guiding users through the purchase process

Table 2 Selection of nudge principles, descriptions, and examples

To define the nudge the goal must be always kept in mind as illustrated in Figure 1. For example, the goal of an ecommerce firm would be to increase the sales. The goal of a tax administration agency would be to encourage citizens to pay taxes and make the process easier. While a project initiator's goal would be to encourage people to make larger donations and pledges [44].

In this thesis feedback principle is going to be incorporated; the user will get nudged to complete their goal, whether they are doing well for the day or not, and how they can easily achieve their daily target.

2.3.2 Understand the user

To understand the users as illustrated in Figure 1 which is the second step in the cycle. Heuristics which also be defined as [44] “rules of thumb” can make decision making process easier by reducing information overload but can also negatively affect the decision making by introducing biases. For example, referring to stereotypes when accessing certain situations. The decision-making process should be understood properly, and goals need to be defined so when designing nudges, the biases are avoided, which may deviate the goal [44].

This step involves understanding the user. The efficiency of the nudge heavily depends on who is being nudged. To have an effective nudge, it is better to get as much information on the user as one can get, so the system can nudge effectively. In this thesis, the user being nudged are inactive users. To effectively nudge, different attributes are to be incorporated into the nudge, such as what activities the user likes, what are the active hours of the user (active hours are *user preferred time to do any kind of activity*) and the goal user is trying to reach.

2.3.3 Design the nudge

To design the nudge as illustrated in Figure 1. When designing a nudge suitable intervention technique can be used as leverage to direct the nudge towards the intended goal. For example, designing the nudge by using the available user interface to preselect the desired option. In case of crowdfunding the goal would be to increase pledges amount so the desired option can be either preselected or made seems attractive than the other available options.[44]

This step involves the implementation or practicality of the nudge. The nudge being developed should comprise of clearly developed goals, understanding of the user, and selecting an appropriate nudge. The intended system developed should implement the nudge through the available user interface (UI). In this thesis, goal of the nudge is to motivate a user to do some activity. The activity will be from pool of activities. The nudge will only happen when the user is in active hours.

2.3.4 Test the Nudge

To test the nudge as illustrated in Figure 1. Different designs can be experimented with when designing a nudge in a digital environment. The effectiveness of nudge can depend on the context, target audience and the intended goal. For example, a digital nudge which works in the context of hotel booking maybe not work in car rental service. Online experiments like A/B and split testing can used to test digital nudges. Through testing only can be found the best available nudge depending on the users and the context which requires the nudge [44].

As the final step in the life cycle, it is important as it will help the nudge architect determine whether the nudge is effective in changing user behaviour or not [44]. The effectiveness of the nudge can be determined by how many users accept the nudge and how many do not. This way, if the nudge is performing poorly, the design architect can retract to step 1 of the digital nudge life cycle and proceed further. In this thesis, the nudge will be tested by real users upon the completion of the system. The results will show the effectiveness of the nudge so it can be improved in future.

2.4 Nudging user

After carefully going through the life cycle of digital nudge, nudging the user defines the working structure of the nudge that will be employed by the intended mobile app. The intended app can be defined as: *The mobile application/user interface that is developed during this thesis.* The aim of the nudge is to trigger a behavioural change in an individual when it comes to physical activity. The main challenge is how to present a nudge to the user, so it has its effect on the user and the user chooses an active and healthy lifestyle. To tackle this challenge, the nudge needs to have knowledge of the user context. Figure 2 shows the attributes of the nudge, which are user goal, activities rated, activities to steps and user activity time.

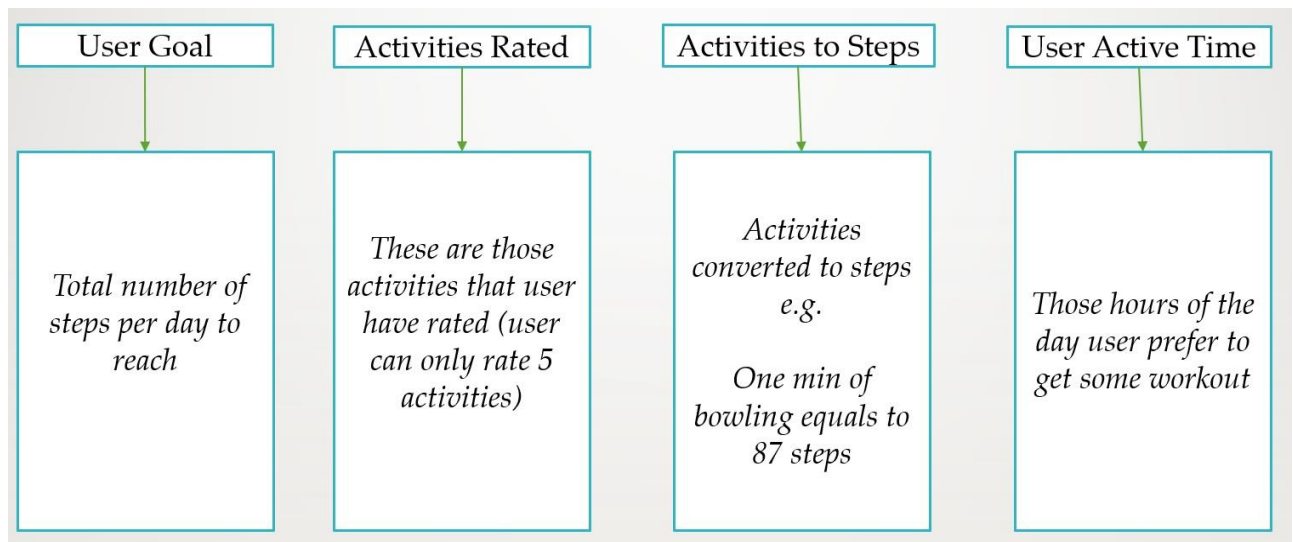


Figure 2 Nudge Attributes

User Goal: The minimum number of steps that the user needs to take on daily basis, as defined by the user. The number of steps need to be greater or equal to 5000 per day.

Activities Rated: These are the activities that the user wishes to do, the user have rated these activities from 0 to 5 depending on how much that specific activity is preferred. The user can rate only 5 activities.

Activities to Steps: These are the user preferred activities, if done for a specific amount of time will be equal to specific number of steps. For example, if bowling is done for one min the number of steps taken will 87. The Activity to steps data has been taken from MIT Get Fit [46].

User Active Time: These are the hours of the day, when the user prefers to get some activity done.

These attributes are the building block of the nudge. Each of these attributes is essential to have an effective nudge. A theoretical working of a nudge consists of the following steps.

- Firstly, user active hours are considered and check if now is a good time to nudge.
- Secondly, if it is time to nudge, it will look at the goal of the user for the day; if the user has already achieved the goal, it will simply present a congratulations message.

- Furthermore, if it's active hours and the user's goal is not yet achieved, the nudge looks at the best possible way to achieve the goal. To do so, the nudge will have to look at the different activities that user prefers and suggest that activity to the user, which consumes less time in reaching the goal. For example, if the user needs 1000 steps to reach the goal, and there are 2 activities user likes. Then the user will be suggested with that activity which equates to more steps per minute. For example, suppose one minute boxing equates to 200 steps/min and one minute running equates to 150 steps/min then the system will recommend the former.
- Lastly, the nudge gives congratulations/feedback upon achieving the goal or encourages the user to do better next time.

3 Recommendation Systems

In today's online world [43], automated recommendations are used as a nudge to recommend users to use and buy services and products. Recommendations are everywhere in the digital world. In online shops, new shopping items are constantly recommended to customers. In Shopify, new songs or artists are suggested, and even the news feed of our social media profile is tailored to the user's preferences. Sellers enhance their sales by carefully reading and analysing user's activities and then suggest us new items, for example Netflix uses this to suggest new titles to customers, and social media platforms suggest new friends; therefore, people can be connected [39]. These recommendations are usually designed to avoid situations of decision overload [43]. In this thesis, recommendation techniques are employed to recommend new activities to users to keep them engaged with an intended mobile application.

When making a recommendation for a particular user, *“the very basic approach is to select the objects favoured by other users that are similar to the target user”* [39]. The intended system will recommend new activities. Nudge, as defined above, means helping people make better decisions without forcing certain outcomes upon them. In this thesis, the recommender system's sole role will be to recommend new activities to the user based on other user-preferred activities. Other user's preferred activities must have one or more similar activity to that of current user. This way, users can enjoy the activities they like along with new activities to explore.

3.1 Types of Recommendation Systems

There are many types of recommendation systems, all of which are according to the different needs of users. All these systems fall into mainly three categories, which are as [47] follows

- **Content-Based Filtering**
- **Collaborative Filtering**
- **Hybrid System**

This thesis will go through the general functionality of the mentioned categories.

3.1.1 Content-Based Filtering

Content-Based Filtering is a method of recommending similar items based on the information of items that the user has selected in the past [47]. Content-Based Filtering does not require other users' profile information. It is highly adaptable as this approach dynamically adapts itself to user-changing preferences. Therefore, it does not depend on other users' data to recommend an item instead; it needs current user information and user's past items [48]. Content-Based Filtering has a huge disadvantage when it comes to recommending new items to the user because it highly depends on previous items, and it can hardly recommend items that the user might not have rated or liked in the past [47]. This type of recommendation [47] is the most basic type of all three categories. Content-Based Filtering was mainly used in e-commerce recommendations, movies recommendation and educational material recommendations [47]. Content-based filtering uses text mining technologies as mentioned by Hyeyoung Ko *et al.* [47].

3.1.2 Collaborative Filtering

Collaborative filtering [49] is, considering the ratings of a user, find all former users who shared the target user's interests based on their ratings, and then giving recommendations about all products that the target user has not yet reviewed/rated. The recommended products are only rated by the target neighbourhood users (same taste to target users). Unlike Content-Based Filtering, whose sole focus is on one user and the items it selected, Collaborative Filtering uses the user's past item taste along with other users that selected similar items and placed them together.

This method finds a group or collection of users X who have similar preferences, likes, and dislikes as user A [48]. In this context, X is called A 's neighbourhood. Users A are then recommended new items that are liked by the majority of users in X [48]. The model efficiency depends on how well it can find the neighbours. This algorithm is further divided into 2 sub-categories, but these categories will be explained further in the thesis.

3.1.3 Collaborative Filtering Challenges

There are some limitations to collaborative filtering. As easy as it is to implement, the challenges of collaborative filtering should be kept in check and tend to those problems that can affect recommendations. Below are some of the challenges that collaborative filtering faces.

3.1.3.1 Data Sparsity

Sparsity [50] problem “*is one of the major problem recommender system faces*” the quality of the recommender system greatly depends on sparsity. Sparsity in data is when there are many items left unrated, meaning the user-item matrix will have many unrated or 0 data, which can have a huge effect on the performance of the recommender. Sparsity can also keep some items from ever suggested/recommended to a user, for example if an item is not rated by any user, that item will not be recommended because of its 0 ratings.

One of the data sparsity problems is cold start. It occurs when a new user or item is introduced to the system. Due to a lack of information, it is challenging to find similar ones (cold start is also referred to as the new user or the new item problem in some publications/literature) [51]. It is not possible to recommend new things until several users have rated them, and since new users lack a rating history, it is doubtful that they will receive favourable recommendations.

3.1.3.2 Synonymy

Synonymy refers to the [51] “*tendency of a number of the same or very similar items to have different names or entries*”. This latent association is not discovered by most recommender systems, and thus these products are treated differently [51]. For example, in the activities list which is obtained from Get Fit MIT [46], there are many activities that are similar but somewhat different according to their intensity. Therefore, traditional collaborative filtering (CF) will fail to find any match between the two and will influence the quality of recommender system. For example, *Bicycling, easy pace* and *Bicycling, vigorous pace* are both similar but traditional collaborative filtering (CF) will treat it differently. Though, in this thesis, this type of difference will also be considered unique rather than the same category. If a user wants to do bicycling at an easy pace, the intended recommender system should also recommend bicycling at a moderate or vigorous pace. Therefore, synonymy will not have much effect on the intended recommender system.

3.1.3.3 Scalability

Scalability [51] is when the number of items and user increases tremendously, the system will suffer serious scalability problems because of computational resources. CF (Collaborative Filtering) heavily depends on computation resources, and when the user-item matrix exceeds millions, the cost of computing similarities between users will be very high.

3.1.4 Hybrid System

The hybrid system, as the name suggests, is the combination or aggregation of two or more techniques for addressing a specific domain. This aggregation can be performed in different ways, it can combine two or more techniques end results and take the most common one. It can also use collaborating filter along with content collaboration for better recommendations. This hybrid [48] incorporation of two or more methods/techniques, many recommender applications generally perform better and are more accurate.

3.1.5 Objective and Recommender systems

Based on the above definitions, we can easily eliminate **Content-Based Filtering** because of the nature of the obtained dataset. As this thesis employ recommendations, it will suggest new activities to the user that they might like; by new activity, it is meant *an activity that the user has not yet tried/rated*, Content-Based Filtering fails at recommending items in this case activity that user have not rated in the past. It leaves us with the other two techniques, Collaborative Filtering and Hybrid Filtering systems, Hybrid Filtering system may have good accuracy and efficient performance, though it also uses content-based filtering along with collaborative filtering, and we have opted out of the possibility of using content-based filtering due to the nature of the data obtained for the intended system.

Due to implicitly getting the data and limited types of activities, content-filtering cannot be utilised because the user will get the same activities from such a limited pool of activities and almost no features/attributes of activities. The more appropriate model to use would be collaborative filtering, and this is because this technique mostly depends on users' ratings. The more users, the better chance of suggesting new activities to target users.

3.2 Collaborative Filtering sub-categories

Collaborating filtering is explained in the above section, but the sub-categories of collaborating filtering will be discussed in this section.

This technique/method has two sub-categories which are as follows:

- **Memory-Based Collaborative Filtering**
- **Model-Based Collaborative Filtering**

3.2.1 Memory-Based Collaborative Filtering

Memory-Based collaborative filtering [48] suggests new items based on the preferences of neighbour users. Memory-Based Filtering is computationally heavy because the model is re-computed every time a new user is added to the system. This model is further divided into two sub-categories, which are [48] user-based collaborative filtering and item-based collaborative filtering.

User-based collaborative filtering approach [48] is to suggest a new item that the target user has not yet rated by calculating the rating of a new item from neighbours' users who have rated this item; if it receives a positive rating, then the item is recommended.

Item-based collaborative filtering [48], a neighbourhood of related items that the user has previously rated is built. The weighted average of all ratings within a neighbourhood of related items is then used to estimate the user's rating for a new similar item.

3.2.2 Model-Based Collaborative Filtering

Model-based collaborative filtering makes use of matrix factorisation techniques [52]. Model-based filtering [48] does not rely on the complete dataset, thus, they are computationally efficient. By complete dataset, it is meant that model-based filtering does not rely on the whole dataset that is obtained from users, rather it uses a subset of the dataset to train, and the remaining portion of the dataset is used to test model performance. The dataset can be divided into 80-20 percent ratio, 80% of the dataset is serving as the training dataset, and 20% of the dataset serving as the testing dataset. This type of system does not require adding new users to the model; thus, re-training of the model is not required. Even so, new users can also get new item suggestions even though they are not part of the model. In this thesis, model-based CF is not employed because, according to the study by Joonseok *et al.* [53], matrix factorisation methods show better performance when the number of users exceeds 3000 (users > 3, 000). In this thesis, the number of users who responded to the survey is 74 in total, which is comparatively very low, which leads to utilising memory-based collaborative filtering instead of model-based.

3.2.3 User-Based collaborative filtering

As discussed in the section 3.2.1, the user-based collaborative filtering model recommends new items to the current user based on preferences of the neighbour. This goes along with the intended recommender system approach of suggesting new activities to new and old users. This approach is aligned with the dataset obtained, and so is item-based collaborating filtering. Both models can perform relatively well, though both models can face difficulty during training when there are no sufficient data. Challenges that collaborative filtering faces are discussed in the section 3.1.3 "collaborative filtering challenges".

The Collaborative Filtering approach assumes that "similar users like similar items" [52]. The collaborative filtering implementation's first step is to obtain the history of users, for example all the activities users rated, and this can be represented as a rating matrix [54]. A rating matrix consists of a table containing rows representing users, columns representing different activities, and a number representing the rating value for each user at the intersection of the rows and columns [54]. In this matrix user's unrated activity will be shown with the absence of a value which can be 0.

Furthermore, after doing the above step, the next is to determine how closely a target user is related to its neighbours. Many techniques can be used to tackle this problem, though [54] the Pearson correlation coefficient is the most popular and has served as a standard for Collaborative Filtering.

Pearson correlation represents similarity by a scale of -1 to +1. A high positive value implies a high correlation, a negative high value suggests an inversely high correlation (when one says True, the other says False), and finally, a correlation of zero indicates uncorrelated items [52].

$$s(u, v) = \frac{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{r}_u)^2 \sum_{i \in \mathcal{I}_{uv}} (r_{vi} - \bar{r}_v)^2}}$$

Figure 3 Pearson correlation

Figure 4 shows the mathematical equation for Pearson correlation [52], “where $s(u, v)$ represents the similarity of users u and v , \mathcal{I}_{uv} represents the set of items rated by both users u and v , r_{ui} and r_{vi} represent the rating of user u or v for item i , and \bar{r}_u and \bar{r}_v represent the mean rating of user u or v , across all items rated”.

Lastly, ratings are calculated for the activities; this rating is computed by a weighted average of the ratings by the neighbours [54]. Now given a user, all their activities would have been rated, and users can be recommended new unexplored top five most rated score, the score which was obtained by doing the above steps. These values can change when a new user is added to the system, the process is repeated, and ratings are assigned to the user's unassigned activities.

The equation that is used to compute a rating estimate for an unrated item by the target user is given in Figure 5

$$\hat{r}_{ui} = \frac{\sum_{v \in \mathcal{N}_i(u)} w_{uv} r_{vi}}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|}$$

Figure 4 Rating prediction (user-based)

“Where $N_i(u)$ represents the number of neighbours that have item i in common with user u , of which v is a particular neighbour user, and w_{uv} is the similarity between the user u and one of its neighbours v ” [52].

To illustrate it theoretically for more clarity, let's suppose there is a user X and Y, and there are only two activities **walking** and **jogging**. Suppose user X gives a high rating to **walking** activity, and user Y give a high rating to **walking** and **jogging** activities. Now upon applying user-based collaborative filtering, user X unrated activity **jogging** will get a rating calculated from neighbour user Y due to the fact that user Y also likes walking. This is just a simple theoretical illustration of how it is supposed to work. The coding implementation of this model is discussed in the Application Design section.

3.2.4 Item-Based collaborative filtering

Item-based collaborative filtering, also known as item-item recommendation [52], *“was first proposed in and thence developed by Amazon.com as part of their in-house Recommender System”* [52]. This collaborative filtering also works as user-based filtering, starting by making a rating matrix; a rating matrix consists of a table containing rows representing users, columns representing different activities, and a number representing the rating value for each user at the intersection of the rows and columns [54]. In this matrix user's unrated activity will be shown with the absence of a value which can be 0.

Furthermore, after obtaining the user matrix, similarity is computed. Unlike user-based, where similarity is calculated between users, in this model/technique, instead of users, the similarity is computed between items. Items, in this case are activities. The method used to calculate the similarity between items is cosine distance. Cosine distance is identified to be the best-suited algorithm for item-based approaches [52].

The cosine distance similarity function compares [52] two samples and calculates how similar they are by analysing the cosine of the angle between them. This scale, which uses +1 and -1 to denote high (direct and inverse) correlation and zero to represent no correlation, is the same as that used in Pearson Correlation.

$$s(i, j) = \frac{\sum_{u \in \mathcal{U}_{ij}} r_{iu} r_{ju}}{\sqrt{\sum_{u \in \mathcal{U}_i} r_{iu}^2 \sum_{u \in \mathcal{U}_j} r_{ju}^2}}$$

Figure 5 Cosine Distance

Figure 6 shows the mathematical equation for Cosine Distance [52] “where $s(i, j)$ represents the similarity of items i and j , across all users that rated such items, r_{iu} and r_{ju} represents the rating of user u for item i or j and \mathcal{U}_{ij} represents the set of common users who rated item i and j ”.

Like user-based CF, a rating for an item (activity) is computed by a weighted average of all accessible ratings in order to generate an estimate of a rating for an unrated item. It is almost identical to user-based rating estimation, though the slight difference is it computes rating by a weighted average of items rated by the target user instead of a weighted average of users rating for a target item.

The equation that is used to compute a rating estimate for an unrated item by the target user is given in Figure 7

$$\hat{r}_{ui} = \frac{\sum_{j \in \mathcal{N}_u(i)} w_{ij} r_{ju}}{\sum_{v \in \mathcal{N}_u(i)} |w_{ij}|}$$

Figure 6 Rating prediction (item-based)

“Where $N_u(i)$ represents the number of neighbours that have user u in common with item i , of which j is a particular neighbour item, and w_{ij} is the similarity between the item i and one of its neighbours j .” [52].

The coding implementation of this model (item-based collaborative filtering) is discussed in section 7.

4 Tools and Technology

Different types of tools and technology were used during this phase to develop a smooth working system.

4.1 Fitbit

Fitbit [55] is an American manufacturer of consumer electronics and fitness equipment. It creates wireless-enabled wearable devices, fitness trackers, smartwatches, pedometers, and associated software. These products also include monitors for heart rate, sleep quality, and the number of steps climbed. Alphabet, Inc. bought the business in 2021. Their mission is to provide you with the motivation and inspiration to be more active and healthier. In order for you to achieve your health and fitness goals, no matter what they may be, we design products and experiences that seamlessly fit into your lifestyle [56]. In this thesis, Fitbit charge 5 is used to keep track of steps taken by the users to make tailored nudges that can help the user get physically active.

4.1.1 Fitbit API's

The Fitbit API enables developers to integrate their Fitbit data into their products, services and applications [57]. Fitbit provides three types of APIs. An application programming interface (API) [58] is a way for two or more computer programs to communicate with each other, and it is not intended to be used directly by the end user.

4.1.1.1 Device and Settings API

This API is only accessible by applications which run on Fitbit devices, which means this API can be used only when developing an application for Fitbit watch or other devices.

4.1.1.2 Companion API

A Companion API [59] provide an additional JavaScript runtime environment within the Fitbit mobile application in order to enhance any Fitbit application. The additional runtime can leverage mobile device resources and sensors, allowing applications to function more efficiently.

The companion API lets the Fitbit device connect to native Android/IOs Applications, which lets the native application communicate with the Fitbit device using the Fitbit application through the companion API. To use this API, the developer must familiarise themselves with

Fitbit studio and create an app which uses companion API. Companion API has lots of limitations, developer can only use Fitbit device properties and cannot use third party API's, it also put restrictions on local and online data storing. This API cannot be used for the development of the intended application, as the intended application has a very complex structure, it will not only store data but also call custom API's for smooth recommendations and nudges. Fortunately, there is another API, web API, which lets developers connect with the user's Fitbit application but not the Fitbit device, and if connecting to the device is not crucial or necessary, web API can be employed, which lets you retrieve all the user data without directly connecting to user's device.

4.1.1.3 Web API

Fitbit makes available [60] to the public web API that developers can use to access Fitbit user data gathered by Fitbit trackers and smartwatches, Aria, and Aria 2 scales, and also log the data manually. In this thesis, Fitbit web API is used, and after thoroughly researching, it was concluded that web API is more beneficial than companion API. For starters, the development phase consumes less time, and there are no direct benefits to actively acquiring Fitbit device data. Additionally, the domain of Fitbit studio is still an unknown territory and needs extensive research and time to develop a companion application.

On the other hand, web API performs all the tasks needed to develop an app that can nudge a user to become physically active, and the intended app can use web API to pull all the required data from the Fitbit device. There are no advantages to using companion API to build the intended application.

4.2 Firebase

Firebase is a Google platform for developing backend applications. Firebase provides [61] tools which can be used by-product developers and marketers to track analytics, report on app crashes and create marking and product experiments. It also enables developers to develop Android, iOS, and web applications. Firebase is trusted by millions of businesses around the world.

Firebase provides numerous services, of which few are:

Analytics: Google Analytics [62] is an app measurement tool, it provides insights into how users engage with apps and how they use them. It is possible to gain a deeper understanding of

the behaviour of users by utilizing analytics data, which will help businesses decide how to market their app and how to improve performance.

Cloud Messaging: Firebase Cloud Messaging (FCM) [63], is a free cross-platform messaging service that allows users to send and receive messages across devices. Users can notify client applications when new email or data is available for syncing by using FCM.

Realtime Database: The Firebase Realtime Database [64], is a cloud-based database. All the stored data is synced across all users in real time, and the user automatically receives updates with the newest data.

Authentication: The Firebase Authentication [65], authenticate users to the any application using backend services, SDKs and UI libraries. In addition to passwords and phone numbers, it also supports authentication using federated identity providers like Google, Facebook, Twitter and others.

Cloud Firestore: Cloud Firestore [66] is a flexible, scalable NoSQL cloud database from Firebase and Google Cloud. It provides tools that can create responsive applications regardless of network latency or Internet connectivity using it as it offers offline support for mobile and web and synchronizes data across client applications in real time with real time listeners. NoSQL [67] *“databases (aka "not only SQL") are non-tabular databases and store data differently than relational tables. NoSQL databases come in a variety of types based on their data model. The main types are document, key-value, wide-column, and graph”*.

Cloud Function: Google Cloud Functions [68], it enables users to connect cloud services and build serverless applications. The cloud function is a single purpose functions that respond to cloud events and the function is triggered only when an event is fired that is being watched.

In this study, two of the firebase services are used; the two services are “Cloud Firestore” and “Cloud Function”. Cloud Firestore is used for storing user data. The actual communication and format of data are discussed in Firestore (section 7.7). The other one is cloud function, which monitors any change in Firestore and calls the Expo server, which is used to send notifications to the intended mobile application.

4.3 React Native

React Native was developed by the Meta platform, which was previously known as Facebook. It is an open-source UI software framework which is used to develop Android, iOS and Web applications [69]. React Native helps developers create a mobile application using web technologies and the app platform's native capabilities. React Native is based on JavaScript. JavaScript is the [70] world's most used programming language for websites; as of 2022, 98% of websites around the globe used JavaScript. React Native ideology is one React, many platforms, it helps developers build cross-platform native applications for Android and iOS operating systems with both sharing the same base code [71]. JavaScript is used in conjunction with React Native to access the platform's APIs, as well as to describe how the application UI appears and behaves using React components, which are bundles of code that can be reused and nested. The React Native platform allows one team to maintain two platforms using the same common technology React, which is a JavaScript library [72].

Developers can use React components to call views from JavaScript, the view is [72] "*the basic building block of UI in Android and iOS development*" even the smallest visual part of an app, such as a text or a button, is a kind of view. As React Native uses the same view as Android and iOS, it creates the corresponding Android and iOS view at runtime. These platform-baked components are known as Native Components. React Native comes with ready to use Native components to start building an app, and these components are called React Native's core components, though there is a limited number of pre-built core components, and developers can build new components using native Android and iOS programming languages.

In this thesis, React Native is used using its Expo framework, so the intended application can reach not only Android users but also iOS users. Though the intended app in this thesis is only developed for Android, in future, that code with little changes can be used to create an iOS app.

4.3.1 Expo

Expo [73] is a set of tools and services built around React Native. It is a framework for building React Native apps, and it lets developer set up and runs their application faster, unlike React Native CLI (command line interface), which requires knowledge of both Android Studio and Xcode for Android and iOS, respectively. To set up and run the application, Expo needs a developer to have JavaScript knowledge and no command over Xcode or Android Studio.

In this thesis, the Expo framework is used because it provides an easy and fast implementation of the intended application.

Framework: In a computer, a framework is [74] a structure that a developer can build software applications on. A framework is often a layered structure indicating what type of apps a developer can build and how they would connect to each other.

4.3.2 Expo Go

Expo Go [75] is a free, open-source program designed to run React Native applications on Android and iOS, without having to build the app locally. In addition to running the projects on the user's owned devices device faster, it also enables the code to be shared with the whole team without requiring any additional code signatures.

How it works: It is *“a native app that is installed on your device. When you run ‘npx expo start’ in your project, Expo CLI starts a development server and generates a QR code. You can then open the Expo Go app on your device and scan the QR code to connect to the dev server”* [75].

4.4 Python

Python [76] is a programming language used to build software, website and do data analysis and automate tasks. Python is a general-purpose language, which means that it may be used to make a wide range of programs and is not specifically designed for any specific use. One of the most popular programming languages today is a result of its beginner-friendliness and versatility. It is ranked as the second-most popular programming language among developers by RedMonk, an industry analysis firm by conducting a survey [76].

Python is used by Netflix to create their recommendation algorithm. It is the language of choice for scientific computing, machine learning, and data science, allowing users to use low-level libraries and clean high-level APIs to boost performance and productivity [77]. Many data science and computing projects prefer to use Python libraries like Pandas, NumPy, and SciPy because they are user-friendly and recommended [77]. As recommender system are also computing problem, this thesis utilises python and its libraries to create recommender model, one of the library used is Pandas.

Pandas: A software package written for python, is used for data analysis and manipulation. In addition to presenting data structures, it also provides methods for manipulating time series data and numerical tables [78].

5 Methods and Methodologies

This section contains detail description of methods and methodologies used in this thesis.

5.1 Research Methods and Methodologies

Two research methodologies are used most frequently throughout a wide range of research types. These two methodologies are quantitative and qualitative research methodologies. While qualitative techniques focus on investigations in an interpretive approach using smaller data sets to produce ideas or develop computer systems or artefacts, quantitative methodologies use experiments and big data sets to achieve a goal [40]. In this thesis, qualitative research approaches were utilised as discussed in section 1.6. It was concluded that user feedback is necessary for system evaluation in order to fully measure the system's effectiveness. Therefore, Interpretivism philosophical assumption is employed by this thesis. Interpretivism is defined by Anne Håkansson [40] as “*Interpretivism assumes that the reality is accessed only via social constructions. They attempt to understand phenomena by exploring richness, depth, and complexity, often in an inductive manner, to discover the meanings people assign to the phenomenon.*”

Research Method: The research method used in this thesis is Non-Experimental research method since the evaluation of the system depends on user’s feedback. Non-Experimental research method is defined by Anne Håkansson [40] as “*The Non-Experimental method describes or predicts behaviour or opinions and can also describe relationships between variables*”. In this method, it is studied how users behave and what they think about the functionalities and interfaces [40]. Research method is defined by Anne Håkansson [40] as “*Research methods provide procedures for accomplishing research tasks. These procedures are how to do the research with initiating, carrying out and completing research tasks*”.

Research Approach: In order to draw conclusions and determine whether something is true or false, research approaches are used [40]. There are two types of research approaches, but this thesis will go through the research approach which is being employed by this thesis, and that research approach is inductive approach (reasoning). Inductive approach is defined by Anne Håkansson [40] as “*it formulates theories and propositions with alternative explanations from observations and patterns. It can also be used in the development of an artifact*”. Based on particular facts, an inductive approach establishes a general proposition [40]. Data is collected

to analysed and gain a better understanding of a phenomenon and determine how it can be viewed differently [40]. As a result of behaviours, opinions, and experiences, an outcome is produced [40]. Therefore, as the presented system evaluation is based on user's opinions this research approach is employed.

5.2 Data collection and Analysis

This section illustrates data collection and analysis method.

5.2.1 Dataset/Data collection

The dataset used in this project/thesis is obtained explicitly, as the goal of the intended recommender system is solely to suggest new activities from the pool of activities to users. The dataset used had activities rated by the user. After intensive research, it was concluded that there was no such dataset publicly available. To tackle this situation, a questionnaire was created that contained 124 different types of activities. The activities were obtained from MIT Get Fit [46]. Get Fit is presented by MIT medical to make the MIT community healthy. The questionnaire was given to multiple individuals, and they were asked to rate the activities they would like to do; they could rate as many activities as they wanted, though a minimum of 5 activities needed to be rated.

The form was then distributed on personal social media platforms and among circle of friends. The responses were collected. 74 individuals filled out the form, and the data obtained was then converted into a clean dataset; by clean, it meant that the activities that were left unrated were removed from the dataset for the purpose of a good recommender system. The dataset was then used in the intended recommender system.

The Questionnaire form was created using a google form. Below is the pattern of the form being used to collect data.

Activities Rating Form



Kindly rate any 5 or more activities. (1 means little interested and 5 means very interested).

This Form data will only be used in a research, kindly fill it with all honesty as the collected data will be used in our model, the data is very crucial to our research and without it, the model can not be built.

I request again, kindly rate at least 5 activities of your choice, and please do not rate activities randomly.

Thank you.

Note: Kindly rate at least 5 activities

 zainulabedin.marwat@gmail.com (not shared) [Switch account](#) 

	1	2	3	4	5
Aerobic Dance Class	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aerobic Fitness Class	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aerobics, low impact	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aerobics, step	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Backpacking	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Badminton (casual to competitive)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ballet	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Baseball	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Basketball (shooting baskets)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 7 Activities Questionnaire

5.2.2 Data Analysis

For analysis following data analysis method is used.

Computational Mathematics: “*Computational Mathematics is used for calculating numerical methods, modelling and simulations with an emphasize on algorithms, numerical methods, and symbolic methods*” [40].

Narrative Analysis: “*Narrative Analysis concerns literary discussion and analysis*” [40] .

Both above-mentioned data analysis methods are utilized in this degree project/thesis. The acquired data was used in recommendation model. Through Computational Mathematics the data was used in mathematical equation as explained in section 3.2, to compute ratings for unrated items and recommend them to users. For testing of the dataset and the recommendation model, Narrative Analysis is utilized, where users were given the recommendations made by the system and asked if the recommendations are relevant to their preference. The analysis showed whether the data acquired was effective in making a working recommendation model and presented which model is performing better in user's opinion.

5.3 Quality Assurance

“Quality assurance is the validation and verification of the research material [40]. According to Anne Håkansson [40] “The qualitative research, with an inductive approach, must apply and discuss validity, dependability, confirmability, transferability and ethics”. As this thesis utilizes inductive approach, the following mentioned quality assurance terms are used.

Ethics: Ethics outline the fact that, participants must be protected, privacy must be maintained, coercion must be avoided, and material must be treated confidentially [40].

Validity: Validity refers to the fact that the research has been conducted in accordance with generally accepted principles and to be sure the results are confirmed and validated by the respondents [40].

Dependability: Dependability refers to the auditing process that provides a means of determining whether conclusions are accurate [40].

Confirmability: Confirmability indicates that the research was conducted without bias or personal evaluation [40].

Transferability: Transferability involves the development of detailed descriptions that can be used by other researchers as a research database [40].

All the above methods were kept in mind during conducting this thesis. For moral obligations participants data was kept private and only the results are discussed in this thesis. The goal of the thesis involves adults between the age of 18-64, disregarding their ethnicity, and educational background etc. Therefore, the participants were chosen randomly for this research.

This research can be replicated, and it is worth noting that the result might slightly differ due to the randomness of the participants because everyone has different personalities and owns different opinions.

5.4 Software development method

Waterfall software development method was used in this thesis. “*Waterfall is a linear method in which there is a big emphasis on collecting requirements and designing the software architecture before doing development and testing*” [79]. This method was employed because it was expensive to involve participants during the development of the whole system, so to avoid that, waterfall method was used. All the essential requirements were gathered and afterwards the system was developed. There were few areas which were highlighted by the participants during system testing that needed improvements, it can be tend to by releasing newer versions.

5.5 Evaluation of the Recommendation model

In recommender systems “*evaluation process can be done using offline analysis, experiments on live users (online), and a combination of them*” [80]. In this thesis, offline analysis/testing which consists of metrics such as RMSE (root mean square error), which is the standard deviation of the residuals (prediction errors) is not implemented. As according to the paper [81] “*online evaluation is most desired, since it can provide accurate results of how good our system is with real users*”, so instead of using offline evaluations, this thesis focuses on using online evaluation or user-based testing.

User-based testing [80] or “*online testing or live user experiments is conducted as an evaluation method for evaluating user performance, satisfaction, participation, and other measurements*”. In this study, user-based testing is performed to evaluate how relevant are the recommendations made by the intended recommender system. To conduct this, a questionnaire is given to a few users upon their consent. They are then requested to select 5 activities from the given activity list and rate them accordingly, and then submit them. Based on the ratings given by the individuals the user-based and item-based recommender presents N number of recommendations, N is probably 5 in this case, but it can be more. These recommendations are forwarded to the respective users, and they are asked for the feedback. The user decides whichever model suggestion is more relevant to their desired/wanted activities. After collecting

the feedback, responses are collected for both recommender systems, and whichever has been selected by more users is used in the final product. The testing phase is done in the Implementation of the System.

6 Requirements and Application Design

This section discusses the requirements and application design of the intended system.

6.1 Requirements

The first step in the system design process is to clarify and document the user's requirements to generate specifications for the system [82]. A requirement is a condition that must be met in order for a user to solve a problem or accomplish an objective [82]. During the development of the intended system no users were involved, and as this thesis is self-made, the requirements of the system were also self-defined. It means there were no third party involved and this is another reason for choosing waterfall software development method.

Following are the requirements for the system.

- The user must rate 5 activities during the registration.
- The user must also set goal greater or equal to 5000 steps per day.
- The user must have a Fitbit watch and Fitbit account to take advantage of the system.
- The user must specify active hours.
- The system must recommend new activities based on user preference.
- The system must nudge user to do suggested activity.
- The system must suggest that activity from the user selected activities, which takes less time to make user reach their daily goal.
- The system must only nudge every hour during user's active hours.
- The system must be secure and easy to use.

6.2 Application Design

The Application design phase focuses on the development of the mobile application, so users can be nudged to perform some activities and not be inactive. This phase consists of Software architecture and UI elements of the system, and it provides a deep overview of the whole application and its working. While designing the mobile application, it was crucial to build the app for both android OS and iOS, and after some planning and research, it was decided that the best way to build the app is in React Native technology instead of creating the same application in a different environment.

React Native is an open-source framework enabling developers to build apps on multiple platforms (different operating systems and hardware) with the same base code. This was a less time-consuming and efficient way to have one baseline code for multiple systems.

6.3 Mobile Application

This main phase objective is to nudge the user along with recommending user-preferred activities. It focuses on the design of the application, which helps the user easily navigate through the system. The system was developed with users kept in mind so users can easily use the system without any special technical knowledge. The application was developed using React Native Expo framework technology.

6.3.1 Cross-platform

The application was developed with the choice of the development platform, which was also influenced by the desire to make the app usable by a variety of user groups. The programme is written in React Native, which enables cross-platform development and deployment to several app stores so as to not limit the customers to a particular operating system.

6.3.2 Connection to Fitbit

Fitbit watch and the mobile app cannot work together directly by connecting through Bluetooth because of company policy, so instead, the intended app will use Fitbit web API to connect to the user's Fitbit account to flawlessly pull steps recorded taken by the user along with heart rate and calories burned, the Fitbit mobile application pulls the data from the user's Fitbit device and store on the server against that particular user, and that is how our intended application is able to pull data from the server using Fitbit web API. In order to make it as easy as possible for users with less technical skills to use the app, only one internal button is created; the button must be touched by the user in order to pair the app with the Fitbit app.

The application is always connected to Fitbit; Fitbit detects the watch automatically and connects to it if it's in pairing mode, and the intended mobile application keeps requesting Fitbit web API for required data.

6.3.3 Sending Notification to Fitbit watch

Besides the step counts that are pulled from web API by the intended app, the other essential part of the application is to send a notification to the user's Fitbit device. During the

development of the application, it was difficult to send a notification directly to the watch from the intended app. Therefore, instead of sending a notification directly to the Fitbit device, the user's Fitbit mobile application is used, this way the Fitbit mobile application catches the notification of the intended mobile app and then forwards it to the user's Fitbit device, this is a two-step process, but it is essential.

6.3.4 Rating activities

One of the other most essential parts of the application is recording user-preferred activities rating. Upon registering user will be asked to rate 5 activities no more or no less. These activities can be rated 1 to 5, whereas 1 means interested but little, and 5 means very interested. This rating of the activities has two essential responsibilities.

- The rating will be used as a baseline for the recommendation, it will give the intended model a taste of user preference, so the user can be recommended new activities that the user might like.
- The other part is choosing an activity that can help the user achieve their goal, for example if a user has already done 4500 steps, and only 500 steps remain, the app will only suggest that activity selected from user-rated activities (5 activities) which by doing user can achieve its target with less effort and can achieve its daily goal.

6.3.5 Recommendation

Besides nudging, the other important part the app does is recommending new activities to the user based on the user preference, the recommendation is given based on the user rating that the user has given to other activities, and if there is any activity that is similar to the user activity, the application will recommend it to the user, the recommendation notification will be displayed on user phone. From the limited set of activities, there can only be limited recommendations.

6.3.6 User Interface

User Interface (UI) is the component of the system that facilitates user interaction with the system in an efficient manner by acting as an intermediary between the user and the system [83]. The user interface is everything that the end user physically interacts with when using the system. Following are the functionality of the application.

6.3.6.1 Login and Register Screen

As soon as the user opens the app, there will be a login screen, and if the user is a new user, then that user has to register and then log in to the application, for already existing users, they can just log in to their account. While new user registers themselves, the user is navigated to the activities selection and rating screen; here, the user must select five activities and rate them, and after doing so, the user is navigated to set up their daily goal. A goal is steps to be taken on each day, and the user can not set a daily goal of fewer than 5000 steps/day. The user is also asked to set up active time. The active time is explained in section 2.4. After doing so, the user is navigated to the home screen.

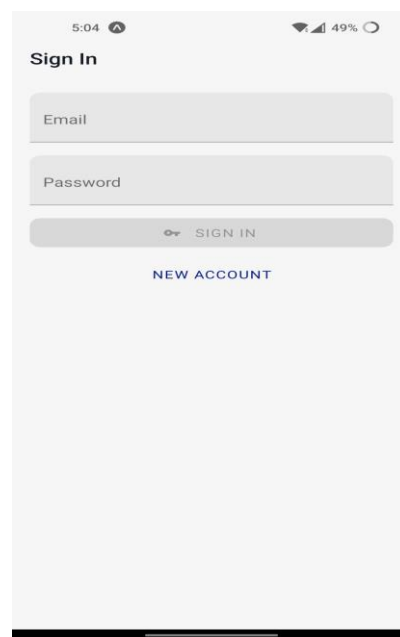


Figure 8 Login Screen

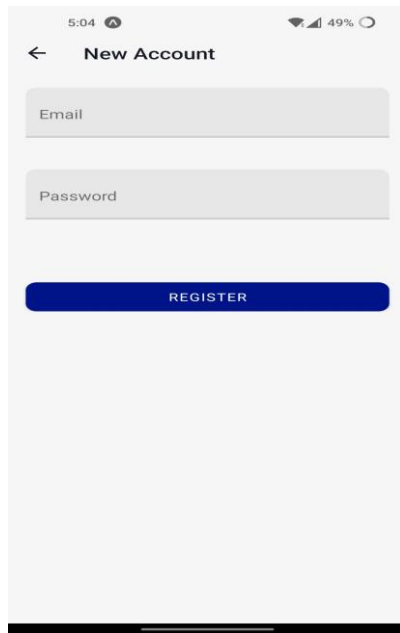


Figure 9 Registration Screen

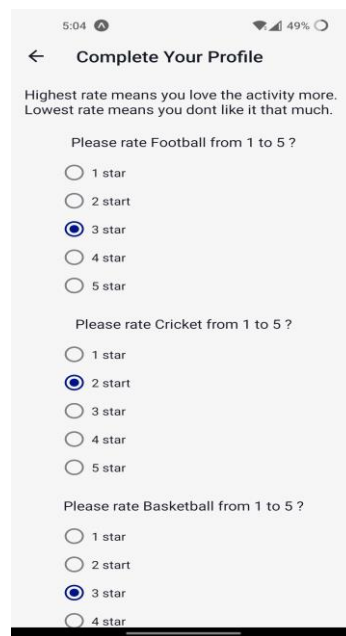
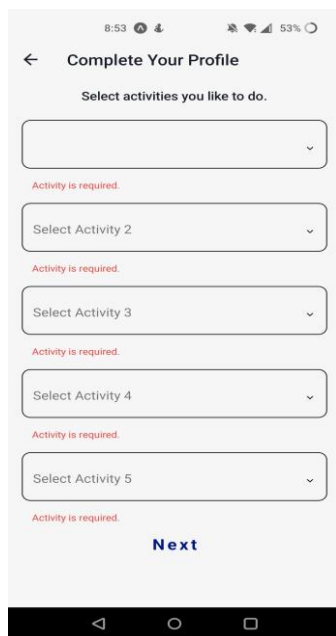


Figure 10 Select and Rate Activities Screen

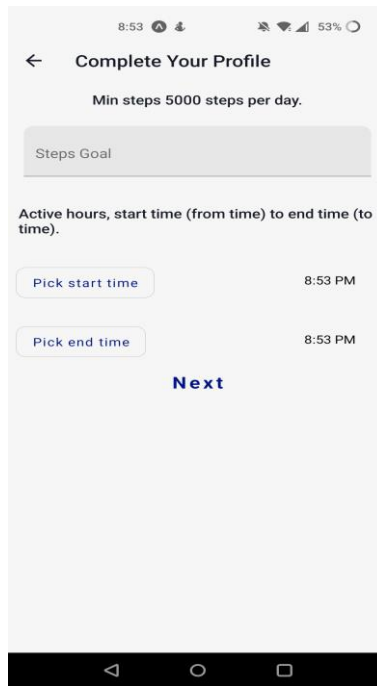


Figure 11 Set-up Daily Goal and Active hours

6.3.6.2 Home Screen

On Home Screen, the user is shown only an option; upon clicking user is asked to sign into their Fitbit account so the app can pull data from the Fitbit server using the Fitbit web API with the user's given credentials. As long as user is logged in, the intended application pulls the data from the user's Fitbit account, processes the data by sending it to Nudge API (section 7.6), which sends a notification to Expo server (section 7.8.1). This notification is then captured by Fitbit mobile application, which sends it to user's Fitbit device.

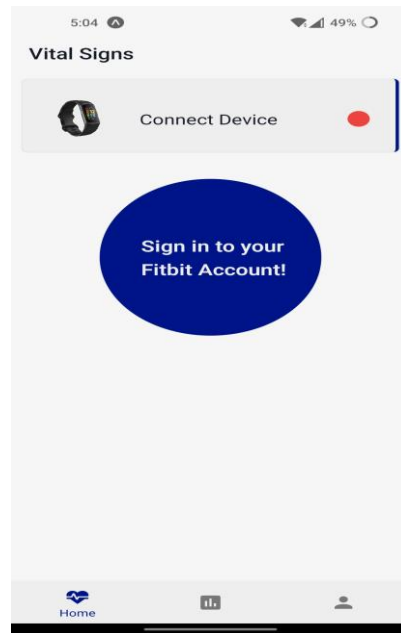


Figure 12 Home Screen

6.3.6.3 Profile Screen and Editing Screen

As the name indicates, this screen presents the user's personal data, and here user can edit the initial activities that the user rated while registering for the application. This is because if along the journey user is recommended an activity by the application, and the user likes that activity, then the user can edit their 5 pre-selected activities and add the recommended activity, then rate it and save the option, though the user can only keep 5 activities at most, so for new activities, user have to delete previous activity.

On the profile screen, the user also has a sign-out option, and upon clicking it, the user is again navigated to the login screen.

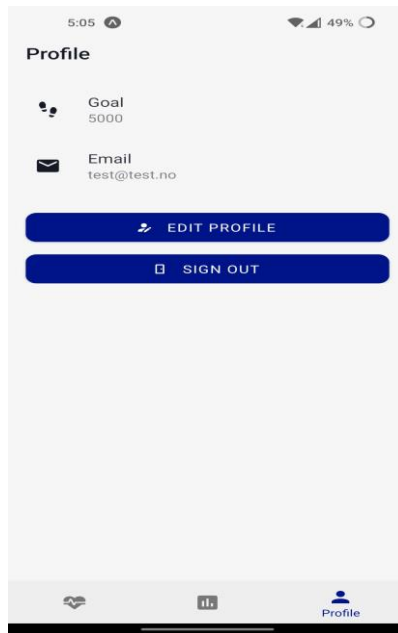


Figure 13 Profile Screen

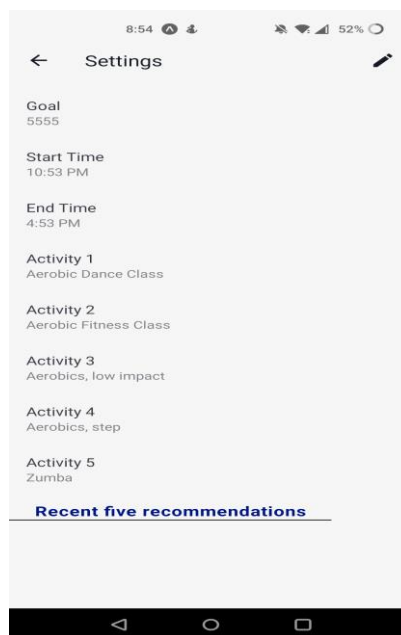


Figure 14 User profile setting screen

The User can then press the pen marker as shown in Figure 14 in the upper right corner of the screen, and after pressing the pen marker user will be taken to the editing screen where the user can not only edit the goal or activity but also edit active hours and activities rating.

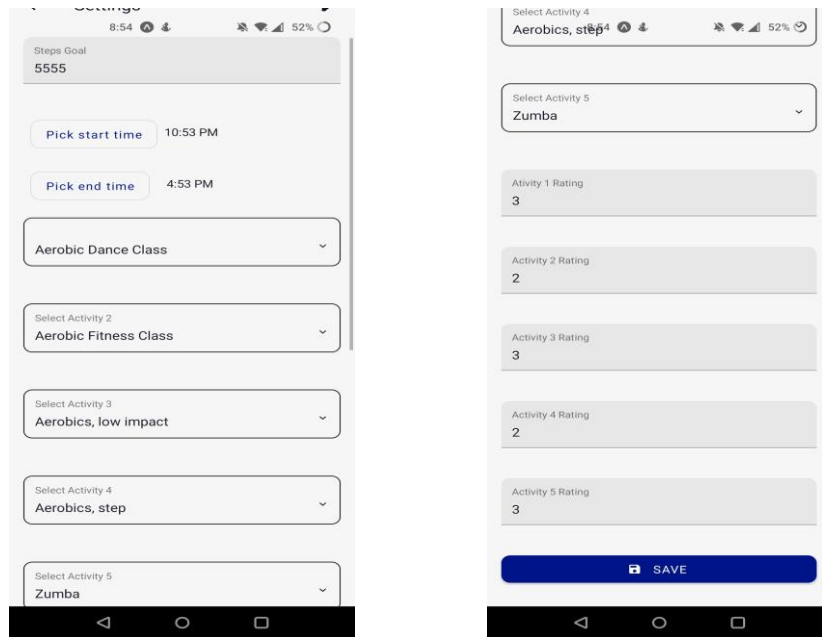


Figure 15 Editing Screen

6.3.7 Intended and Fitbit App

With the newly developed application, Fitbit mobile app should not have been needed, though it is clearly stated in section 6.3.6.2 that a new user has to login into a Fitbit account instead of connecting to the Fitbit device directly, it is because, as observed in section 4.1.1.2 that to directly connect to Fitbit device, companion API is to be employed and to do so another application needs to be developed that runs inside Fitbit mobile application and let the native application communicate with the Fitbit device. In section 4.1.1.3, it is argued that web API is more suitable, less time and resource-consuming.

This led us to the development of a separate mobile application that uses the user's Fitbit account to acquire their data using Fitbit API with the user's credentials instead of the user's device and sends notifications and nudges to the user's Fitbit device through their Fitbit mobile application.

6.3.8 Sources of Data

Dataset used was obtained using a questionnaire, as discussed in section 5.2.1. Other sources of data that are gathered and used by the intended app are from the Fitbit device, GetFit and the user.

Fitbit Device: The steps of the user, the heartbeat and calories of the user are continuously acquired by the intended app during the active hours of the user, and the data is used in producing a nudge. Fitbit device steps are used to check whether the user has accomplished the goal. The steps are also used to decide which activity should be suggested to the user.

GetFit MIT Medical: The activity data used in this thesis is based on Get Fit by MIT Medical. GetFit provided activities to steps conversion, which shows if an individual does an activity for a minute, then that would be equivalent to N number of steps, where N is any number, such as one minute of bowling is equivalent to 87 steps. To make a nudge more acceptable, activities to step are crucial because the aim is to suggest an activity which takes a minimum amount of time for a user to reach their goal.

User Data: The last source of data is the user itself. To make a user profile, user data is a must. The user is asked to select 5 activities and rate them so new activities can be recommended. User is also asked to set a goal and to give their Fitbit account data, so the intended application is able to get data from the user device using Fitbit API.

7 Implementation of the System

This section of the thesis contains the implementation of the whole system, the implementation of the recommendation system, the development of API, storing data in a database and communication of the system through a different set of APIs.

7.1 Implementation of User-Based Recommendation System

The programming language that is used to implement the recommender system is python, as discussed in section 4.4. Python is used by Netflix for their recommender system and is recommended choice for computational problems, and as it can be observed throughout the thesis that recommender models are computational problems.

The basic algorithmic structure follows the mathematical equation of user-based collaborative filtering. Two libraries were used to carry out the whole process, one of the library/modules (a library is a collection of reusable code that is already written by experts) is Pandas (section 4.4), and the other one is SciPy. SciPy provides many different algorithms and functionalities, though, during the development of the recommender model, only Pearson correlation (section 3.2.3) functionality is adopted.

```
1 userRatedActivities = {}
2 for item in items:
3     for user in users:
4         if df[item][user] == 0:
5             allUsersRatedItem = df[df[item] > 0][item]
6             simUsers = allUsersRatedItem.index.tolist()
7             ratings = []
8             similarity = []
9
10            for simUser in simUsers:
11                similarity.append(pearsonr(df.T[user], df.T[simUser]))
12                ratings.append(df[item][simUser])
13
14            similarity = [sim[0] for sim in similarity]
15            df[item][user] = sum([similarity[i]*ratings[i] for i in range(len(ratings))])/sum([abs(sim)
16                                                                                             for sim in similarity])
17        else:
18            if user not in userRatedActivities.keys():
19                userRatedActivities[user] = [item]
20            else:
21                userRatedActivities[user].append(item)
22
```

Figure 16 User-Based Implementation

Figure 16 shows the implementation of user-based collaborative filtering, and Line 11 shows the use of the *pearsonr* function obtained from the SciPy library. This function returns the similarity between two users. It can be any value between -1 and 1 inclusive, *df* at lines 4, 5

etc., which is the object of Pandas library, which contains the user-item DataFrame. It is a 2-dimensional structure with labels.

Figure 16 code snippet computes the rating for the unrated items by users, which can then be used to recommend new activities to users. After the computation 2-dimensional structure (DataFrame) is obtained, the data is then uploaded to firestore. Firestore is defined in section 4.2.

7.2 Implementation of Item Based Recommendation System

The concept, environment and programming language used for the development of Item-Based collaborative filtering is the same as discussed in Implementation of User Based Recommendation System (section 7.1). The only difference is that instead of a user-based collaborative filtering mathematical equation, Item-based collaborative filtering was used, the equation for this is given in section 3.2.4. Instead of using *pearsonr* from the SciPy library, cosine similarity was used.

The application computes user ratings for unrated activities, which can then be used to suggest new activities to users. After computation, the data is then uploaded to the firestore.

7.3 Evaluation of the Model

In section 5.5, it was argued that online testing/experiment is more desirable as it is tested by real users. In this thesis, online testing is performed to determine whether user based is good or item-based collaborative filtering. To carry out online testing, a questionnaire was prepared. It was similar to the questionnaire used in obtaining the dataset (section 5.2.1) for the model. This questionnaire required user email, so afterwards, when recommendations are made, they can be emailed to each individual.

There were 17 individuals that took part in this experiment. Each individual selected 5 activities and rated them based on their liking. Out of 17, 3 individuals were ostracised because they rated more than 15 activities which made them outliers. Out of the remaining 14, 11 individuals responded. The participants after selecting the activities they liked and rated, each individual data was given to both recommender systems to make recommendations. Both user-based and item-based collaborative filtering recommended 5 activities each, which were then sent to each participant by their provided email in a carefully written email.

11 participants out of 17 responded, 6 participants liked the suggestions/recommendations made by item-based collaborative filtering, and the other 5 participants gave credit to user-based collaborative filtering. Based on this real user experience/responses, it was concluded that item-based collaborative filtering is better and is used in this thesis's intended application.

Greetings,

Hope you are doing well, first of all I thank you for participating in this survey, it means a lot to my thesis, without further ado, I present to you the recommendations made by my two models. Kindly after careful consideration, suggest the model that you think recommended the most relevant activities.

User-Based Recommendations:

Walking (moderate)
Backpacking
Bicycling, easy pace
Hiking (general)
Fishing

Item_Based Recommendations:

Wash car
Tae Kwon Do
Scrub Floors
Fencing
Running (10mph, 6 minute/mile)

Above **User-Based** and **Item-Based** recommendations are 2 different models, recommending activities that you might like based on your pre-selected activities.

Kindly after deciding which model is performing/recommending relevant activities, email me back with the name of the model you think is good.

I will really appreciate your collaboration.

Thank you
regards
Sufyan Saleem

Figure 17 Email Format

7.4 Recommendation System API

To make recommendations for the desired users, the intended mobile application uses API to talk with the recommendation system. Both systems are separate entities, and they cannot directly interact with each other. The API of the system is developed using Flask Framework. Flask is web-framework in python.

The intended application calls the recommendation API, passing user ID and steps as parameters to the API, and in return, API returns a JSON object. A JSON (JavaScript Object Notation) is a standard for text-based data representation and is based on JavaScript object syntax [84]. It is used for sending some data from the server to the client and vice-versa. The following syntax shows the return format of the data from recommendation API.

```

{
  "Activities": [
    {
      "activity": "Walking (moderate)",
      "id": 1
    },
    {
      "activity": "Bicycling, leisurely",
      "id": 2
    },
    {
      "activity": "Backpacking",
      "id": 3
    },
    {
      "activity": "Bicycling, easy pace",
      "id": 4
    },
    {
      "activity": "Fishing",
      "id": 5
    }
  ]
}

```

In the above syntax, *Activities* is a property that contains an array of objects. Each object then has two more properties which are *id* and *activity*. The API not only returns the data to the mobile application but also uploads the data to the firestore.

7.5 Nudge Script

The nudge script is a python written application/script that processes data that is sent from the intended mobile application. The mobile application sends the user Id and user steps taken till the request to the script. The script then calls firestore using firebase admin SDK. Firebase admin SDK [85], provides libraries that allow interaction with Firebase services from a privileged environment. For example, admin has the ability to read and write real-time data at any time. [86]. The script pulls user's goal (user daily steps target) along with all the activities users have rated from the firestore using Firebase admin SDK.

After pulling the required data, it is processed, sorted, and send back to the nudge API, which sends it back to the requested party, which in this case is the intended application.

7.6 Nudging API

Although the mobile application does the nudging, the data for the nudge is obtained from the nudge API. The intended mobile application calls the nudge API, passing the user Id and user steps taken at the time of calling the API. The API then process the required data and sends back information which is used in nudging. The nudge API returns the following data in JSON format.

```
{  
  "TimeToReachGoal": 9,  
  "activity": "kickboxing"  
}
```

In the above syntax, *TimeToReachGoal* and *activity* both are properties that have 9 and *kickboxing* as values, what it means is that the intended user should do a *kickboxing* activity for 9 minutes to reach their goal. The data explained here is real data gathered from a user during the testing of the nudging API.

When the intended application sends the user Id and the steps the user has taken till then, the nudge API requests for processing of the data being sent (by the user/intended application), the data is sent to a separate python script, which then does the processing of computing and suggesting the activity and the time it takes to reach the goal, the script defined is mentioned in section 7.5.

7.7 Firestore

In Firestore, data is [66] “*stored in documents that contain fields mapping to values. These documents are stored in collections, which are containers for your documents that you can use to organise your data and build queries. Documents support many different data types, from simple strings and numbers, to complex, nested objects. You can also create subcollections within documents and build hierarchical data structures that scale as your database grows. The Cloud Firestore data model supports whatever data structure works best for your app*”. The Firestore for the intended application has four collections. Each collection has a set of

documents. These documents contain fields mapping to values. To get a clear understanding of how data is stored and accessed, each collection is explained thoroughly.

1. Collection profiles
2. Collection ratingDataset
3. Collection recommendations
4. Collection users

Collection profiles: This collection stores the users' token/id as a document. Each document represents one user, it has fields, and each field has some value or values. The following data represents data inside one user/document.

Activities:

```
activity1: "Kayaking"
activity2: "Sailing"
activity3: "Shopping (mall)"
activity4: "Aerobics, step"
activity5: "Handball"
```

email: "lsalalito@gmail.com"

endTime: October 3, 2022 at 9:20:33 PM UTC+5

goal: "4500"

ratings:

```
question1: "5"
question2: "2"
question3: "4"
question4: "5"
question5: "4"
```

startTime: October 3, 2022 at 4:20:33 PM UTC+5

timestamp: October 3, 2022 at 5:21:01 PM UTC+5

userId: "6azl9NOZZJRnKcnQMKzPfUOquL43"

From the representative data, every value written on the left side of the colon is a field, and the value on the right side of the colon is a value in a document. The data shown is for the user having Id *6azl9NOZZJRnKcnQMKzPfUOquL43*. Fields can have subsets of different Fields. In the above example, *ratings* and *Activities* contains a collection/set of other fields which have specific values.

Collection ratingDataset: This collection has only two documents, which are *ratingOriginalDataset* and *ratingComputedDatasetItems* both of these documents contain the users' Id as a field, and all the 124 activities as a set of fields, and each activity field has a rating as value. The following demonstrated structure gives a glimpse of how the data is structured.

```
User Id:  
  Football: 0  
  Boxing: 4  
  Kickboxing: 5  
  Walking: 3  
  ....
```

The only difference between the two documents is that *ratingOriginalDataset* contains actual ratings given by the user, and *ratingComputedDatasetItems* contains ratings computed by item-based collaborative filtering.

Collection recommendations: This collection also has a user id/token as a document, and each document contains *Activities* as a field which contains an array of fields mapping values. These fields are 5 activities recommended by item-based collaborative filtering.

Collection users: This collection contains general information about users. It also contains the user id/token as a document, which contains the user's information as fields mapping values. The following example gives an illustration of how data is stored.

```
Email: "zslayer05@gmail.com"  
expoToken": ExponentPushToken[D5vrodKEPySvyNdxYOUppK]"  
registered: true  
timestamp: October 11, 2022 at 8:38:29 PM UTC+5  
userId: "DOJGU32LCyPxSjpFQ1Hzj0wks2F2"
```

7.8 Cloud Function

Firebase Cloud Function is [68] a "*serverless execution environment for building and connecting cloud services*". The developer can create a function that monitors Firebase Firestore for any changes and notify the client application that new email or data is available. It provides the user Expo token which is stored in Collection Users (section 7.7) against the target user id/token. Expo token is created when the user registers on the intended mobile

application, which is stored in Collection Users (section 7.7). Cloud function provides the user token along with other data (explained in section 7.8.1 Expo Push notification) to Expo Push notification and so the user gets a notification. Notification messages can be sent to encourage user re-engagement and retention. Cloud function in the intended system is implemented such that it gets triggered whenever a user/document is edited, deleted, or added in Recommendation Collection (section 7.7). The function then gets the user Expo token value which triggered the cloud function from *Users Collection* and sends a notification to the user using that expo push notification. The expo push notification is captured by the Fitbit mobile application (only if the intended app is enabled to send notification in the Fitbit mobile application settings) which forwards it to the user's Fitbit device.

How Nudging user for an activity works: As the working of cloud function is explained, it can be observed that the cloud function only gets triggered when there is a change in *recommendation collection* and not gets triggered when user have not reached the goal. Therefore, to nudge user to do some activity instead of cloud function the Expo API (section 7.8.1) is called directly from the intended application, the intended application first calls the Nudging API (section 7.6), which returns requested data, that data is then sent to Expo API and user get a notification stating, *Do Boxing for 30 minutes*.

7.8.1 Expo Push Notification

Expo Push Notifications [87] are an important feature, providing users with information about new releases, sales, deals, or messages from friends, push notifications can also be used to increase user interaction and create a better overall user experience by informing them of things they may be interested in.

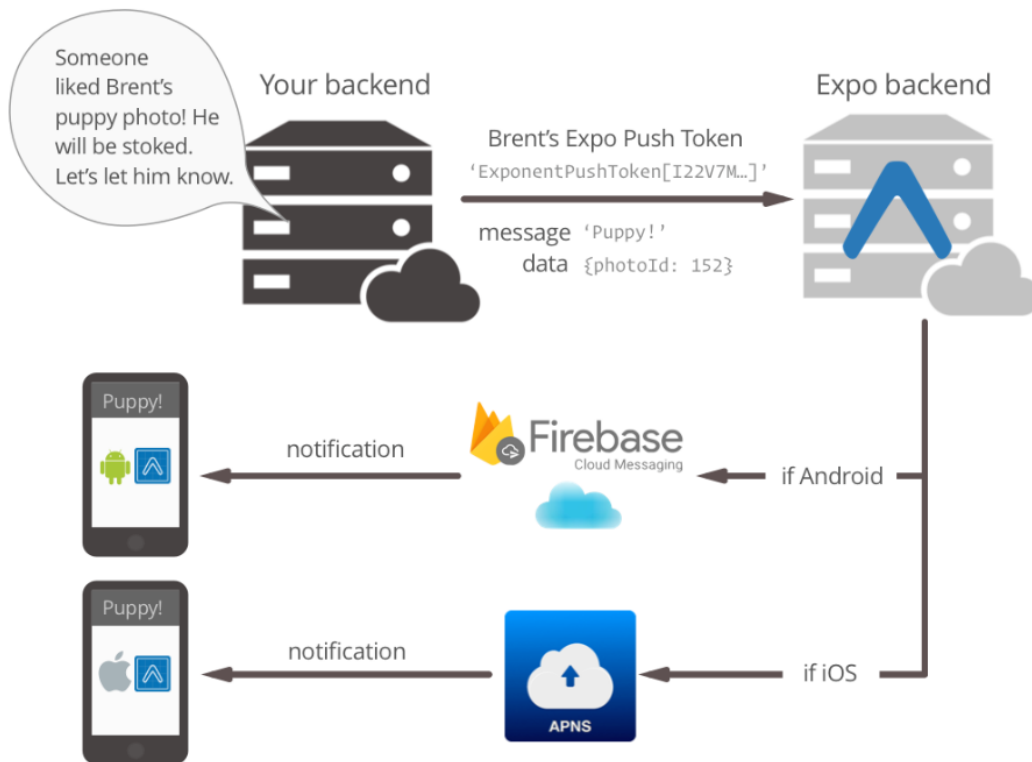


Figure 18 Expo Push Notification

Figure 18, which is taken from [88], shows the flow of the push notification. The cloud function looks at the *recommendation document* and gets triggered if anything is changed in the document. It then notifies the expo backend, as shown in Figure 18, which sends a notification directly to the intended application. Although, Figure 18 shows the expo backend sends message to *Firebase cloud messaging* which then sends it to android application or uses *APNS* to do the same for iOS applications, but in this thesis, both *cloud messaging* and *APNS* are not employed as the intended application is running on Expo server using Expo go (section 4.3.2) mobile application, which directly receives a notification as it is already running on Expo server.

How it works: "When you're ready to send a push notification, take the Expo push token from your user record and send it to the Expo API using a plain old *HTTPS POST* request. To do so, send a *POST* request to <https://exp.host/--/api/v2/push/send> with the following *HTTP* headers" [88]. Therefore, when the cloud function (section 7.8) gets triggered, Expo API is called using

`https://exp.host/--/api/v2/push/send` URL. The data being sent to the Expo API has the following structure [88].

```
curl -H "Content-Type: application/json" -X POST "https://exp.host/--  
/api/v2/push/send" -d '{  
  "to": "ExponentPushToken[xxxxxxxxxxxxxxxxxxxxxx] (User Expo token)",  
  "title": "Congratulations",  
  "body": "You have reached the goal."  
}'
```

The above code snippet is cuRI POST request to the Expo API, the body of the request is a JSON Object. The contents of the request body are, to which includes the Expo push token indicating the user that will receive the notification, while the title and the body mentioned is what the user receives and is displayed on the user's operating system when receiving the notification from Expo.

7.9 Flow of the whole system

This section illustrates the overall flow of the system, from user creation to recommendations and nudges. This section will go through each step in detail, and every point is a scenario that demonstrates the working of the intended application.

- First of all, a user registers themselves on the intended mobile application. They are asked for the five activities they would like to do and their ratings. They are also asked for their daily goal (which cannot be less than 5000 steps) and their active hours. Section 2.4 explains active hours in detail. In short active hours represents those hours of the day in which the user prefers to do some activity or to be nudged for some activity.
- After registration, the user logs in to their Fitbit account, so the intended application keeps track of user steps, as explained in sections 6.3.7 and 6.3.8.
- The intended mobile application continuously pings Firestore API for active hours of the user, which is stored in *profile collection* if it is active hours, the intended application immediately calls recommendation API.
- During the start of the active hours of the user, the intended application first calls the recommendation API (section 7.4) by passing the user Id, which is obtained from the *profile collection* the recommendation API updates the *recommendation collection* with

5 activities recommended by Item-Based collaboration filtering, upon doing so triggers cloud function (section 7.8) which is continuously monitoring the *recommendation collection*. Cloud function, as explained in section 7.8, sends a notification to the user, which says, *check your settings of the intended mobile application for recommended activities*.

- Furthermore, the intended application calls the Nudge API (section 7.6) after calling the recommendation API, this time intended application calls the Nudge API after every hour (till the end of active hours). It sends the user Id and steps taken till the request and the Nudge API sends back the activity to do and how much time it will take for the user to cover the remaining steps, then the application nudges the user with the return values stating, *kindly do this activity for x minutes to reach your goal for the day*, if the user has already achieved the daily goal, the application will not nudge the user. How the nudge/notification is sent to the user Fitbit device is explained in section 6.3.3
- If the user is not logged in, the intended application will not send any notifications to the user.

8 System Testing

To test the intended mobile application, real users were invited to participate in the testing of the mobile application. This test was not only intended to check if the mobile application works, but it also focused on the impact of the idea, which is reducing physical inactivity in inactive users. There were 6 participants involved, only few of them had their own Fitbit device, the other was provided the Fitbit device given by the university. Each of them was given detailed instructions on how to install the intended application and what its purpose is. All the participants were instructed to check the intended mobile application for few days and give their feedback on bugs in the application, areas to improve and its impact on their daily life.

The aim of involving real users was to check the validity of the thesis's proposed solution to reduce inactivity. With real users, it is possible to not only test the application but also judge the validity of the idea that it has any impact on individuals. The intention was to involve large number of participants, but due to lack of Fitbit device, many couldn't get involved, while others were not willing to participate, that is why the test size is small.

8.1 Results

All of the responses/feedback from the volunteer pointed out that the application is working fine, and it has areas to improve. Few of them even had an impact on their daily activity routine. The following points were gathered from volunteer feedback.

- Four of the volunteers said the application has an impact on their daily life. They claimed that the app kept them engaged in doing activities. One of them even said, *the application's gentle reminder created some sort of guilt which kept me going. If it was not for the app, I would have remained inactive.* The other 2 participants' feedback lacked information about the application's impact.
- The respondents pointed out that the app should have a historical record to keep track of our daily achievements.
- One of the volunteers suggested it would be motivational if the application had the capability of showing calories burned, average heart rate and daily calorie intake.
- It was noticed that there was a lack of appreciation for completing a goal. One of the participants even pointed out that upon completing the goal, the application should reward the individual with some sense of achievement, like nice words etc.

It seems that overall feedback on the system was positive. It can be observed that the application indeed had an impact on the user's life. Though the test body was small, out of 6, 4 individuals/participants had an impact on their life, which suggests the proposed idea of reducing physical inactivity might help change people's lives, though, to actually prove this hypothesis, a large test size would be required, with a lot of resources such as Fitbit devices etc.

9 Future Work and Conclusion

To provide excellent quality of user experience, there are many features that can be added to the overall system. Few of the suggested features by the participants were already meant to be part of the intended application, though, due to limited resources and data, they were excluded.

Below are the features/screens that can be added to the mobile application.

9.1 Overview Screen

Here user can see steps taken on the current day, and the user can also see real-time heart rate chart, calories and steps taken charts. The chart is of two types, User can either check their daily heart rate, calories etc., or the user has the option to see a weekly chart, this chart will show the average heart rate on the selected day of the week. It can also show the average steps taken or calories burned on a particular day of the week.

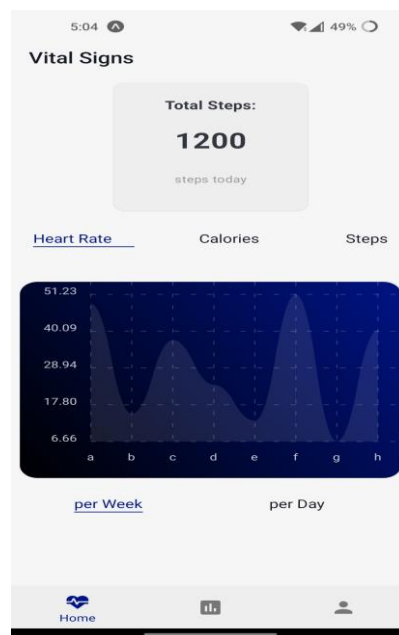


Figure 19 Overview Screen

9.2 Historical Data Screen

On this screen, user can check their activities of the past. On this screen, there is a calendar that the user can access and select a specific day, for which that day record will be shown to the user. By default, all the records are shown to the user. Each record represents a day, and each record has three values, the date on which the record was recorded, the steps the user has taken on that day, and whether the user achieved that day's goal. This way, users can easily keep track of their progress and see daily, weekly, or monthly progress.

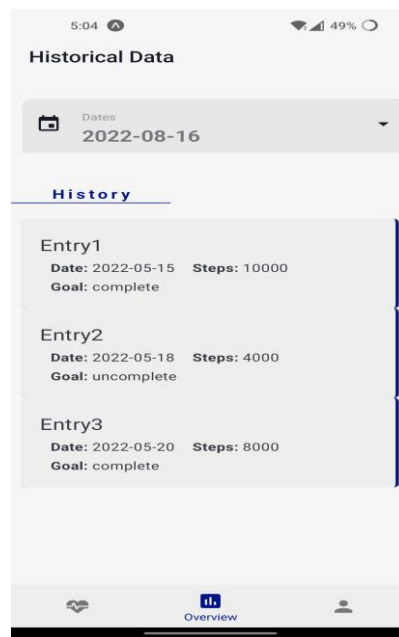


Figure 20 Historical Data Screen

Above are some of the features that can be added to the mobile application. These are also the features that were suggested by the participants/volunteers. The mobile application only works with the Fitbit watch, which makes its impact limited. In future, the application can be made to accept any kind of smart watch which have some predefined qualities. This will let the application reach numerous numbers of people. The application also lacks feedback capabilities. The feedback on recommended activities would not only help the developer improve the recommendation, but it will also suggest that the user has accepted or rejected a nudge. One of the participants suggested that the application needs to be more engaging and fun to use because after some time people will be unwilling to try/use the application anymore, to tackle this problem, initial idea was to create a system that not only nudge a user for an

activity which takes less time but also to suggest to user an activity based on multiple factors, for example weather, distance to the activity, the user age, the transport system, users stress level (which can be get from the Fitbit device) and many more, this way user can be more motivated to do an activity and would be willing to use the app regularly.

The main purpose of the thesis was to create a digital nudge that can make inactive users active. To improve its precision and acceptance rate, the application can include many contextual information which can help make a more personalised smart nudge, for example, if current weather, distance and time to reach the activity location, age, height and gender of user etc., can help make even more personalised nudges, which can make acceptance rate of nudges even higher, even more, the nudge can be made more adaptive to user's daily routine, to do so a lot of time and effort is needed and in future along with application the nudge can be improved and it can have an impact on thousands of people.

9.3 Conclusion

In this thesis, an approach for nudging and recommender systems has been presented. The purpose of the application is try to reduce inactivity among inactive individuals, the focus has been on making a personalised nudge which can tempt individuals to not be inactive, and recommendations were also introduced to make it more engageable. The application seems to work well with users, and it can tempt the user to do some activity. This thesis presented a solution to a problem, and it worked for the majority of the user that participated in the testing of the system and its impact.

Works cited

- [1] “NUDGING | meaning in the Cambridge English Dictionary.”
<https://dictionary.cambridge.org/dictionary/english/nudging> (accessed Mar. 02, 2022).
- [2] M. G. Hunnes, “Nudging”.
- [3] E. Selinger and K. Whyte, “Is There a Right Way to Nudge? The Practice and Ethics of Choice Architecture,” *Sociol Compass*, vol. 5, no. 10, pp. 923–935, Oct. 2011, doi: 10.1111/J.1751-9020.2011.00413.X.
- [4] T. M. Marteau, D. Ogilvie, M. Roland, M. Suhrcke, and M. P. Kelly, “Judging nudging: can nudging improve population health?,” *BMJ*, vol. 342, no. 7791, pp. 263–265, Jan. 2011, doi: 10.1136/BMJ.D228.
- [5] R. Karlsen and A. Andersen, “Recommendations with a Nudge,” *Technologies 2019*, Vol. 7, Page 45, vol. 7, no. 2, p. 45, Jun. 2019, doi: 10.3390/TECHNOLOGIES7020045.
- [6] H. W. Kohl *et al.*, “The pandemic of physical inactivity: global action for public health,” *The Lancet*, vol. 380, no. 9838, pp. 294–305, Jul. 2012, doi: 10.1016/S0140-6736(12)60898-8.
- [7] I. M. Lee *et al.*, “Effect of physical inactivity on major non-communicable diseases worldwide: an analysis of burden of disease and life expectancy,” *Lancet*, vol. 380, no. 9838, pp. 219–229, 2012, doi: 10.1016/S0140-6736(12)61031-9.
- [8] Y. Liang, “Recommender System for Developing New Preferences and Goals,” *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019, doi: 10.1145/3298689.
- [9] L. Palokangas Lauri and J. Suomala JyrkiSuomala, “Nudging Problematic Smartphone Use to a Lower Level”.

- [10] T. C. Leonard, “Richard H. Thaler, Cass R. Sunstein, Nudge: Improving decisions about health, wealth, and happiness,” *Constitutional Political Economy* 2008 19:4, vol. 19, no. 4, pp. 356–360, Aug. 2008, doi: 10.1007/S10602-008-9056-2.
- [11] M. Tremblay, “Letter to the editor: standardized use of the terms ‘sedentary’ and ‘sedentary behaviours,’” *Appl Physiol Nutr Metab*, vol. 37, no. 3, pp. 540–542, Jun. 2012, doi: 10.1139/H2012-024.
- [12] R. L. Newton, H. Han, T. Zderic, and M. Hamilton, “The Energy Expenditure of Sedentary Behavior: A Whole Room Calorimeter Study,” *PLoS One*, vol. 8, no. 5, p. e63171, May 2013, doi: 10.1371/JOURNAL.PONE.0063171.
- [13] J. Y. Chau, H. P. van der Ploeg, D. Merom, T. Chey, and A. E. Bauman, “Cross-sectional associations between occupational and leisure-time sitting, physical activity and obesity in working adults,” *Prev Med (Baltim)*, vol. 54, no. 3–4, pp. 195–200, Mar. 2012, doi: 10.1016/J.YPMED.2011.12.020.
- [14] H. Du *et al.*, “Physical activity and sedentary leisure time and their associations with BMI, waist circumference, and percentage body fat in 0.5 million adults: the China Kadoorie Biobank study 1-3,” *Am J Clin Nutr*, vol. 97, pp. 487–96, 2013, doi: 10.3945/ajcn.112.046854.
- [15] J. Henson *et al.*, “Associations of objectively measured sedentary behaviour and physical activity with markers of cardiometabolic health,” *Diabetologia*, vol. 56, no. 5, pp. 1012–1020, May 2013, doi: 10.1007/S00125-013-2845-9/FIGURES/2.
- [16] A. Bankoski *et al.*, “Sedentary Activity Associated With Metabolic Syndrome Independent of Physical Activity,” *Diabetes Care*, vol. 34, no. 2, pp. 497–503, Feb. 2011, doi: 10.2337/DC10-0987.
- [17] C. E. Matthews *et al.*, “Amount of time spent in sedentary behaviors and cause-specific mortality in US adults,” *Am J Clin Nutr*, vol. 95, no. 2, pp. 437–445, Feb. 2012, doi: 10.3945/AJCN.111.019620.
- [18] L. M. León-Muñoz, D. Martínez-Gómez, T. Balboa-Castillo, E. López-García, P. Guallar-Castillón, and F. Rodríguez-Artalejo, “Continued sedentariness, change in

- sitting time, and mortality in older adults,” *Med Sci Sports Exerc*, vol. 45, no. 8, pp. 1501–1507, Aug. 2013, doi: 10.1249/MSS.0B013E3182897E87.
- [19] D. W. Dunstan *et al.*, “Breaking Up Prolonged Sitting Reduces Postprandial Glucose and Insulin Responses,” *Diabetes Care*, vol. 35, no. 5, pp. 976–983, May 2012, doi: 10.2337/DC11-1931.
- [20] J. W. van Dijk, M. Venema, W. van Mechelen, C. D. A. Stehouwer, F. Hartgens, and L. J. C. van Loon, “Effect of Moderate-Intensity Exercise Versus Activities of Daily Living on 24-Hour Blood Glucose Homeostasis in Male Patients With Type 2 Diabetes,” *Diabetes Care*, vol. 36, no. 11, pp. 3448–3453, Nov. 2013, doi: 10.2337/DC12-2620.
- [21] A. M. Swartz, L. Squires, and S. J. Strath, “Energy expenditure of interruptions to sedentary behavior,” *International Journal of Behavioral Nutrition and Physical Activity*, vol. 8, no. 1, pp. 1–7, Jun. 2011, doi: 10.1186/1479-5868-8-69/FIGURES/2.
- [22] C. Latouche *et al.*, “Effects of breaking up prolonged sitting on skeletal muscle gene expression,” *J Appl Physiol*, vol. 114, no. 4, pp. 453–460, Feb. 2013, doi: 10.1152/JAPPLPHYSIOL.00978.2012/ASSET/IMAGES/LARGE/ZDG0041304740003.JPEG.
- [23] G. N. Healy *et al.*, “Reducing sitting time in office workers: Short-term efficacy of a multicomponent intervention,” *Prev Med (Baltim)*, vol. 57, no. 1, pp. 43–48, Jul. 2013, doi: 10.1016/J.YPMED.2013.04.004.
- [24] S. Kozey-Keadle *et al.*, “Changes in sedentary time and physical activity in response to an exercise training and/or lifestyle intervention,” *J Phys Act Health*, vol. 11, no. 7, pp. 1324–1333, Sep. 2014, doi: 10.1123/JPAH.2012-0340.
- [25] D. S. Bond *et al.*, “B-MOBILE - A Smartphone-Based Intervention to Reduce Sedentary Time in Overweight/Obese Individuals: A Within-Subjects Experimental Trial,” *PLoS One*, vol. 9, no. 6, p. e100821, Jun. 2014, doi: 10.1371/JOURNAL.PONE.0100821.

- [26] D. S. Bond, J. G. Thomas, J. L. Unick, H. A. Raynor, S. Vithiananthan, and R. R. Wing, "Self-reported and objectively measured sedentary behavior in bariatric surgery candidates," *Surgery for Obesity and Related Diseases*, vol. 9, no. 1, pp. 123–128, Jan. 2013, doi: 10.1016/J.SOARD.2012.09.008.
- [27] T. Scheers, R. Philippaerts, and J. Lefevre, "SenseWear-determined physical activity and sedentary behavior and metabolic syndrome," *Med Sci Sports Exerc*, vol. 45, no. 3, pp. 481–489, Mar. 2013, doi: 10.1249/MSS.0B013E31827563BA.
- [28] M. Papastergiou, "Exploring the potential of computer and video games for health and physical education," *Comput Educ*, vol. 53, no. 3, pp. 603–622, Nov. 2009, doi: 10.1016/J.COMPEDU.2009.04.001.
- [29] T. Campbell, B. Ngo, and J. Fogarty, "Game Design Principles in Everyday Fitness Applications," *Proceedings of the ACM 2008 conference on Computer supported cooperative work - CSCW '08*, 2008, doi: 10.1145/1460563.
- [30] Z. Zhao, A. Arya, R. Orji, and G. Chan, "Physical Activity Recommendation for Exergame Player Modeling using Machine Learning Approach," *2020 IEEE 8th International Conference on Serious Games and Applications for Health, SeGAH 2020*, Aug. 2020, doi: 10.1109/SEGAH49190.2020.9201820.
- [31] A. Sultana, S. Tasnim, M. M. Hossain, S. Bhattacharya, and N. Purohit, "Digital screen time during the COVID-19 pandemic: a public health concern," *F1000Res*, vol. 10, pp. 1–8, 2021, doi: 10.12688/F1000RESEARCH.50880.1.
- [32] P. C. Hallal *et al.*, "Global physical activity levels: surveillance progress, pitfalls, and prospects," *The Lancet*, vol. 380, no. 9838, pp. 247–257, Jul. 2012, doi: 10.1016/S0140-6736(12)60646-1.
- [33] C. Tudor-Locke *et al.*, "How many steps/day are enough? for adults," *International Journal of Behavioral Nutrition and Physical Activity*, vol. 8, no. 1, pp. 1–17, Jul. 2011, doi: 10.1186/1479-5868-8-79/FIGURES/1.

- [34] D. J. Macfarlane, D. Chan, K. L. Chan, E. Y. K. Ho, and C. C. Y. Lee, "Using three objective criteria to examine pedometer guidelines for free-living individuals," *Eur J Appl Physiol*, vol. 104, no. 3, pp. 435–444, 2008, doi: 10.1007/S00421-008-0789-4.
- [35] "Evidence Behind 10,000 Steps Walking | Journal of Health Research." <https://he01.tci-thaijo.org/index.php/jhealthres/article/view/89095> (accessed May 04, 2022).
- [36] C. Tudor-Locke, Y. Hatano, R. P. Pangrazi, and M. Kang, "Revisiting 'how many steps are enough?,'" *Med Sci Sports Exerc*, vol. 40, no. 7 SUPPL.1, 2008, doi: 10.1249/MSS.0B013E31817C7133.
- [37] "Physical Activity Recommendations for Different Age Groups | Physical Activity | DNPAO | CDC." <https://www.cdc.gov/physicalactivity/basics/age-chart.html> (accessed May 17, 2022).
- [38] Y. Zhao and B. Shen, "Empirical Study of User Preferences Based on Rating Data of Movies," 2016, doi: 10.1371/journal.pone.0146541.
- [39] L. Lü, M. Medo, C. H. Yeung, Y. C. Zhang, Z. K. Zhang, and T. Zhou, "Recommender Systems," *Phys Rep*, vol. 519, no. 1, pp. 1–49, Feb. 2012, doi: 10.1016/j.physrep.2012.02.006.
- [40] H. R. Arabnia, Nev. International Conference on Frontiers in Education: Computer Science & Computer Engineering 2013.07.22-25 Las Vegas, Nev. FECS 2013.07.22-25 Las Vegas, and C. E. World Congress in Computer Science, "Portal of Research Methods and Methodologies for Research Projects and Degree Projects," *Computer Engineering, and Applied Computing WORLDCOMP*, pp. 67–73, 2013, Accessed: Aug. 16, 2022. [Online]. Available: <http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-136960>
- [41] S. Dalecke and R. Karlsen, "Designing Dynamic and Personalized Nudges," 2020, doi: 10.1145/3405962.3405975.

- [42] C. Mele, T. Russo Spena, V. Kaartemo, and M. L. Marzullo, “Smart nudging: How cognitive technologies enable choice architectures for value co-creation,” *J Bus Res*, vol. 129, pp. 949–960, May 2021, doi: 10.1016/J.JBUSRES.2020.09.004.
- [43] M. Jesse and D. Jannach, “Digital nudging with recommender systems: Survey and future directions,” *Computers in Human Behavior Reports*, vol. 3, p. 100052, Jan. 2021, doi: 10.1016/J.CHBR.2020.100052.
- [44] C. Schneider, M. Weinmann, and J. vom Brocke, “Digital nudging: Guiding online user choices through interface design Designers can create designs that nudge users toward the most desirable option,” *Commun ACM*, vol. 61, no. 7, pp. 67–73, Jul. 2018, doi: 10.1145/3213765.
- [45] M. Weinmann, C. Schneider, and J. vom Brocke, “Digital Nudging,” *Business and Information Systems Engineering*, vol. 58, no. 6, pp. 433–436, Dec. 2016, doi: 10.1007/S12599-016-0453-1.
- [46] M. Get Fit, “Convert Activities to Steps.” Accessed: Aug. 25, 2022. [Online]. Available: https://getfit.mit.edu/sites/default/files/resource/Convert_activities_to_steps.pdf
- [47] H. Ko, S. Lee, Y. Park, and A. Choi, “A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields,” *Electronics 2022, Vol. 11, Page 141*, vol. 11, no. 1, p. 141, Jan. 2022, doi: 10.3390/ELECTRONICS11010141.
- [48] D. Roy and M. Dutta, “A systematic review and research perspective on recommender systems,” *J Big Data*, vol. 9, no. 1, pp. 1–36, Dec. 2022, doi: 10.1186/S40537-022-00592-5/TABLES/3.
- [49] A. Lokesh, “A Comparative Study of Recommendation Systems,” *Masters Theses & Specialist Projects*, Oct. 2019, Accessed: Jun. 25, 2022. [Online]. Available: <https://digitalcommons.wku.edu/theses/3166>

- [50] H. Al-Bashiri, M. A. Abdulgaber, A. Romli, and F. Hujainah, "Collaborative filtering recommender system: Overview and challenges," *Adv Sci Lett*, vol. 23, no. 9, pp. 9045–9049, Sep. 2017, doi: 10.1166/ASL.2017.10020.
- [51] X. Su and T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," *Advances in Artificial Intelligence*, vol. 2009, pp. 1–19, Oct. 2009, doi: 10.1155/2009/421425.
- [52] C. Levinas, "An Analysis of Memory Based Collaborative Filtering Recommender Systems with Improvement Proposals," *undefined*, 2014.
- [53] J. Lee, M. Sun, and G. Lebanon, "A Comparative Study of Collaborative Filtering Algorithms," *KDIR 2012 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*, pp. 132–137, May 2012, doi: 10.48550/arxiv.1205.3193.
- [54] Z.-D. Zhao, M. S. Shang, and M.-S. Shang, "User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop," 2010, doi: 10.1109/WKDD.2010.54.
- [55] "Fitbit - Wikipedia." <https://en.wikipedia.org/wiki/Fitbit> (accessed Sep. 17, 2022).
- [56] "About Fitbit." <https://staticcs.fitbit.com/about> (accessed Sep. 17, 2022).
- [57] "Fitbit." <https://www.fitbit.com/dev> (accessed Sep. 17, 2022).
- [58] "API - Wikipedia." <https://en.wikipedia.org/wiki/API> (accessed Sep. 17, 2022).
- [59] "Companion Guide." <https://dev.fitbit.com/build/guides/companion/> (accessed Sep. 17, 2022).
- [60] "Web API." <https://dev.fitbit.com/build/reference/web-api/> (accessed Sep. 17, 2022).
- [61] "What is Google Firebase? - Definition from WhatIs.com." <https://www.techtarget.com/searchmobilecomputing/definition/Google-Firebase> (accessed Sep. 17, 2022).

- [62] “Google Analytics for Firebase.” <https://firebase.google.com/docs/analytics> (accessed Sep. 17, 2022).
- [63] “Firebase Cloud Messaging.” <https://firebase.google.com/docs/cloud-messaging> (accessed Sep. 17, 2022).
- [64] “Firebase Realtime Database.” <https://firebase.google.com/docs/database> (accessed Sep. 17, 2022).
- [65] “Firebase Authentication.” <https://firebase.google.com/docs/auth> (accessed Sep. 17, 2022).
- [66] “Firestore | Firebase.” <https://firebase.google.com/docs/firestore> (accessed Oct. 11, 2022).
- [67] “What Is NoSQL? NoSQL Databases Explained | MongoDB.” <https://www.mongodb.com/nosql-explained> (accessed Oct. 11, 2022).
- [68] “Cloud Functions overview | Cloud Functions Documentation | Google Cloud.” <https://cloud.google.com/functions/docs/concepts/overview> (accessed Oct. 23, 2022).
- [69] “React Native - Wikipedia.” https://en.wikipedia.org/wiki/React_Native (accessed Nov. 14, 2022).
- [70] “JavaScript - Wikipedia.” <https://en.wikipedia.org/wiki/JavaScript> (accessed Sep. 16, 2022).
- [71] “React Native · Learn once, write anywhere.” <https://reactnative.dev/> (accessed Nov. 14, 2022).
- [72] “Core Components and Native Components · React Native.” <https://reactnative.dev/docs/intro-react-native-components> (accessed Sep. 17, 2022).
- [73] “Setting up the development environment · React Native.” <https://reactnative.dev/docs/environment-setup> (accessed Sep. 17, 2022).

- [74] “What is framework? - Definition from WhatIs.com.”
<https://www.techtarget.com/whatis/definition/framework> (accessed Sep. 17, 2022).
- [75] “Expo Go - Expo Documentation.” <https://docs.expo.dev/workflow/expo-go/> (accessed Nov. 01, 2022).
- [76] “What Is Python Used For? A Beginner’s Guide | Coursera.”
<https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python> (accessed Sep. 17, 2022).
- [77] “(PDF) Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence.”
https://www.researchgate.net/publication/340457442_Machine_Learning_in_Python_Main_Developments_and_Technology_Trends_in_Data_Science_Machine_Learning_and_Artificial_Intelligence (accessed Sep. 16, 2022).
- [78] “pandas (software) - Wikipedia.” [https://en.wikipedia.org/wiki/Pandas_\(software\)](https://en.wikipedia.org/wiki/Pandas_(software))
(accessed Sep. 17, 2022).
- [79] “(PDF) Software Development Methodologies.”
https://www.researchgate.net/publication/255710396_Software_Development_Methodologies (accessed Nov. 12, 2022).
- [80] P. H. Aditya, I. Budi, and Q. Munajat, “A comparative analysis of memory-based and model-based collaborative filtering on the implementation of recommender system for E-commerce in Indonesia: A case study PT X,” *2016 International Conference on Advanced Computer Science and Information Systems, ICACISIS 2016*, pp. 303–308, Mar. 2017, doi: 10.1109/ICACISIS.2016.7872755.
- [81] T. Silveira, M. Zhang, X. Lin, Y. Liu, and S. Ma, “How good your recommender system is? A survey on evaluations in recommendation,” *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 5, pp. 813–831, May 2019, doi: 10.1007/S13042-017-0762-9/TABLES/4.

- [82] “Paper for Topic: Requirements & Specifications.”
https://users.ece.cmu.edu/~koopman/des_s99/requirements_specs/ (accessed Nov. 12, 2022).
- [83] “(PDF) User Interface Design Issues for Easy and Efficient Human Computer Interaction: An Explanatory Approach.”
https://www.researchgate.net/publication/294428623_User_Interface_Design_Issues_for_Easy_and_Efficient_Human_Computer_Interaction_An_Explanatory_Approach (accessed Aug. 14, 2022).
- [84] “Working with JSON - Learn web development | MDN.”
<https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON> (accessed Oct. 11, 2022).
- [85] “SDKs and client libraries | Firestore | Firebase.”
<https://firebase.google.com/docs/firestore/client/libraries> (accessed Oct. 11, 2022).
- [86] “Add the Firebase Admin SDK to your server.”
<https://firebase.google.com/docs/admin/setup> (accessed Oct. 11, 2022).
- [87] “Push Notifications Overview - Expo Documentation.” <https://docs.expo.dev/push-notifications/overview/> (accessed Oct. 23, 2022).
- [88] “Sending Notifications with Expo’s Push API - Expo Documentation.”
<https://docs.expo.dev/push-notifications/sending-notifications/> (accessed Nov. 02, 2022).

