Integrating gamification to modern web fitness services

Chenyu Wei

University of Tampere
Faculty of Natural Sciences
Software Development
M. Sc. thesis
Supervisor: Timo Nummenmaa
July 2017

University of Tampere
Faculty of Natural Sciences
Software Development
Chenyu Wei: Integrating gamification to modern web fitness services
M.Sc. thesis, 64 pages
July 2017

Abstract

Gamification has been widely applied to information technologies industry, especially to fields where motivation and user engagement are difficult to maintain. Thus, many fitness providers have integrated gamification to their own services.

However, due to lacking unified gamification convention in Fitness convention, choosing what game-like elements to integrate seems like an impossible task. In addition, there is no valid guideline available for integrating gamification to web fitness application for the industry to follow.

Aiming at these questions this thesis summarizes thirteen most commonly used game-like elements in different web fitness services, and generates guidelines for implementing gamified web fitness services. Results from validating the guidelines to three example fitness services indicate promising percentages of gamification integration, although future work on the topic would significantly increase the validity of the guidelines.

Keywords: gamification, gamified services, fitness, motivation, web services, guidelines.

Acknowledgements

First and foremost, I would like to thank Timo Nummenmaa for supervising this master thesis, who has provided tremendous guidance and helped me overcome many obstacles. I would also like to express my gratitude to Zheying Zhang for her expert advice throughout my writing progress.

I would like to thank Michael Bernstein, Anastasia Kozina, Yokochi Hironori, and Timo Kivikangas for collaborating with the Trexplore project, which has inspired the creation of this thesis topic.

Over the years of study life in Tampere, I have met many Chinese friends who share similar interests and value; here I would like to particularly thank Ye He, Yiwen Li, Pengfei Lv, Chenlu Wang, Yanzhao Wen, Mengling Xu, Xiaolan Ma, Geting Zhu, and Qianqian Zhang for their wonderful friendships. In addition, my special thanks go to my good friends, Wenyan Yang, Dan Yang, and Gaoming Zheng, who have put up with being roommates with me.

It has been a long journey studying in Finland, I would love to thank Renfei Zhou for his extraordinary friendship for the past 8 years. I could not imagine how different my life would become if he were not around in both my bachelor and master studies. Finally, I would like to thank my family for their unconditional love and trust, who have always supported my decision no matter what. Even though 7400 kilometers are quite distant, the gap between our hearts is never existent.

Contents

1. Introduction

Gamification has become a phenomenon methodology in building information systems in the recent decade [Hamari, 2015]. In this area, many scholars have provided the definition for gamification based on their own conceptual development, which sometimes confuse [Deterding et al., 2011; Huotari and Hamari, 2012; Marczewski, 2012]. Gamification have been applied in many fields, especially in the fields which provide services for education or learning, health or exercise, work, intra-organization, and innovation or ideation [Hamari et al., 2014].

Meanwhile, mobile devices have been widely equipped for sport-tracking functionality (for example with helps of pedometer, gyroscope, GPS); in addition, various wearable devices such as fitness wristbands and sport tracking earbuds have increased burgeoning popularity. Thus, there are a lot of services which appeal to integrate gamification to their sport-tracking systems, as gamification provides positive results on the healthy outcomes and costs of services, where it has been identified to enhance an individual's fun, engagement and compliance in accomplishing fitness activities [Lenihan, 2012].

1.1 Current problem

The integration of gamification to fitness system have been proved effective, according to the research by Lister et al. [2014]. However, with chaotic definitions of gamification, it is hard to identify appropriate gamification that are utilized in modern web fitness services.

In addition, even though burgeoning use of *gamification components* and *game elements* have been identified in health and fitness applications, standard guidelines for integrating these elements have been neglected by the industry [Lister et al., 2014].

1.2 Research questions and method

Targeting on the problems presented in the previous section, this thesis addresses the following two research questions:

1. How is gamification utilized in modern web fitness services?
2. What are the best practices for implementing fitness web services which integrate gamification elements?

This thesis will analyze the current problem existed in the literature, and provide a modified methodology for gamification in fitness services. In addition, this thesis will study the current fitness web services and categorize them into two main categories. Based on the literature review

and improved methodology, this thesis will generate guidelines for solving the second research question.

1.3 Structure of the Thesis

As the work of this thesis contains two key parts: gamification and fitness web services, there are several separate chapters introducing these two concepts, in order to help readers who are unfamiliar with these concepts to have a better understanding of the research areas.

There are seven chapters in this thesis. Chapter 1 introduces the background of this research, the problems unsolved from current literature, the research questions raised for this thesis, and the structure introduction for this thesis. Chapter 2 presents the theoretical background of game and play, motivation, engagement and flow, which are the foundation of gamification theory. Chapter 3 presents the current studies of gamification, including different definitions of gamification provided by different scholars, reviews of gamification, as well as the process for gamifying a system, several examples of gamified systems, criticism and risk of gamification, and lastly the current studies of gamification in fitness services. Chapter 4 contains the introduction of web services and their architectural concepts, the commonly used REST APIs and securing methods for websites and mobile applications, and the architecture of wearable devices. Chapter 5 categorizes two main fitness web services, together with their corresponding architecture construction, gamification suggestion, advantages and limitations; Chapter 5 also outlines the guidelines for integrating gamification to modern web fitness services by several implemented examples. Chapter 6 discusses the results of this research, and finally, Chapter 7 draws the conclusion, in addition to suggesting future work for this topic.

2. Theoretical background

This chapter introduces the theoretical concepts that fundamentally facilitate gamification. The first section presents and discusses general definition of games and play. The second section explains the concept of motivation, and how do different motivation theories influence modern software developing mindset. The third section elaborates the definition of engagement and flow.

2.1. Games and play

Throughout the history of human race, games have been played for over thousands of years. There is evidence that for decades, philosophers and wisemen have been trying to define the meaning of games and reason for playing games. The first modern attempt can be traced back to the publication of *Homo Ludens* by Huizinga [1938]. In this innovative book, Huizinga describes playfulness as a built-in personality of human race and a necessary and meaningful condition for the human culture.

Huizinga's work has influenced and been carried out by numerous subsequent scholars, notably by Caillois [1958] who emphasizes the central role of play in the human culture. Caillois systematically classifies the form of play and games, given the considerable difficulty in defining play, he has concluded six essential characteristics for an activity to become a play:

- It is free, playing is not obligatory. Otherwise, it would immediately kill the fun and attractive attribute as diversion.
- It is separate, time and space are predefined prior to play, which totally differs from the basic routine of life.
- It is uncertain, the result from the play cannot be determined beforehand, which makes player's initiative an important factor for the output.
- It is unproductive, no extra goods nor wealth is produced, which makes it unstained during the process of play.
- It is governed by rules, the play is under convention by predefined conditions and constrains, regardless of ordinary laws or behaviors.
- It is make-believe, the existence of the imagined reality is set to be outstanding from the real life, in where players are believed to play.

In Addition, Caillois also raises four main rubrics for classification of games. First, *Agôn*, as known as competitive games, in which equality of chances is artificially created and players contest according to their inner abilities and skills, eventually leads to the winner's triumph;

Second, *Alea*, in contrast to *agôn*, includes those games based on chance where aleatory events dominate; Third, *mimicry*, or role playing, summarizes games where one can escape themselves and become another. Last, *ilinx* or *vertigo*, meaning whirlpool in Greek, or the sense of rapidly location altering movement, which would cause a state of disorder and dizziness.

Many widely-played games can be categorized into one of above classified elements. For instance, game of chess is an *agôn game* and playing slot machines is purely *alea*. However, games can combine those elements in various way to increase the complexity and playfulness. For example, dancing is purely an *ilinx* activity, but *mimicry* can be combined in order to perform role-playing show; similarly, *agôn* can be identified in terms of dancing competition.

The works of Caillois and Huizinga have made a great foundation in the field of game and play theory. However, some would argue that with the rise of pro gaming and play for pay, those who play for a specific goal do not fit in neither of the six characteristics of games. Therefore, carries on from previous studies, Bernard Suits introduces the psychological attitude that required for playing a game, namely the lusory attitude [1978]. He also presents the additional definition for a player entering into the play of a game, which is "the voluntary attempt to overcome unnecessary obstacles". A more detailed definition is also included:

> "To play a game is to attempt to achieve a specific state of affairs, using only means permitted by rules, where the rules prohibit use of more efficient in favor of less efficient means, and where the rules are accepted just because they make possible such activity."

That is to say, even though in a scenario where prelusory goal is heavily weighed for players to achieve, the lusory goal needs to be presented in order that the game be played [Tamminen, 2015]. For example, in a professional tennis series, the prelusory goal is to win all other opponents and get the rewards, but without the ulterior passion for tennis, and lusory goal of enjoying the match, the game would soon become meaningless and not counted as a play in the end.

The studies of play have been carried on by Sutton-Smith, who states that the diversity of play forms and experiences can be illustrated as various if larger menagerie of the play sphere is taken consideration. Nearly anything can allow play to occur within given boundaries [Sutton-Smith, 1997]. However, it would bring up some chaotic ambiguity, due to the lack of coherence in the play theory. Hence, in the publication of *The Ambiguity of Play*, he presents the rhetorical solution for underlying various play theories and play terms. Play ideological theories based on seven distinct rhetorics have been suggested, where the play's definition is broad enough to include all kinds of passive and vicarious forms; thus, this definition should be universally accepted and applied no matter what is the player's age, race, or even species. The seven rhetorics

are elaborated below, consist of first four ancient rhetorics and latter three modern rhetorics:

- Rhetorics of Fate: As the most widespread rhetorics and the first rhetorics from the ancient group, this normally refers to plays depending on probability and randomness, as well as supposition of destiny overruling human lives.
- Rhetorics of Power: Long enrooted in history, this rhetoric has been existing as old as patriarchy. Usually associated with contests or matches, the attendant of the play struggle for proving superiority among other players, in order to bring fulfillment, achieve glory, or seek compensation.
- Rhetorics of Identity, where play is served as a means of recognitive and representative social identities expressing in parades, celebrations and other community based mass spectacles.
- Rhetorics of Frivolity, where play is oppositional, parodic and sometimes revolutionary. Based on the archetypes of the trickster and the fool, this rhetoric refers to the playful activities oppose against the usual beliefs in social and cultural order of daily lives.
- Rhetorics of Progress: This rhetoric notes children adapt and develop usefully through play. Hence, the developmental aspects of play are inevitable in human society.
- Rhetorics of the Imaginary: The collective rhetorics where imagination, creativity, flexibility and innovation are the most fundamental characteristic of play.
- Rhetorics of the Self: Focus on play as serving its basis of individual playful pursuits and interests, this rhetoric is more concerned with individuals than with groups, that play is considered as a form of escaping daily life flow and seeking fun from freedom.

Those rhetorics described above creates numerous ways of ambiguous, consequently tremendous ambiguous playful activities in the world are able to find their rhetorics respectively. For example, Juveniles contributing role-playing games reflects the rhetoric of progress, gamblers relying luck in chance games reflects the rhetoric of fate, actors imposing creativity and flexibility in fantasy plays reflects the rhetoric of imaginary.

In recent decades, evolutional technologies have been utilized in game industry: video games based on personal computers have been widely played, in which the contents of games and rules of play have become more and more complicated, so the theoretical theories mentioned above need some expansion and redefinition. Thus, notably in their monumental book *Rules of Play: Game Design Fundamentals* [2003], Katie Salen and Eric Zimmerman describe play as free movement in a rigid structure. In addition, they also state that all games have rules to be learned and obeyed for players, although majorities of the players pursuing the fun and lusory in the

progress of play, some players attempt to break the rules if had chance, to acquire the easy achievement of winning. Hence, means of cheating avoidance should be introduced, either by allowing a judge to interfere in traditional games, or a strict and robust anti-cheating system in modern digital games [Engeli, 2004].

Besides certitude and obedience of rules, players also need to invest some certain amount of time and effort in order to perform play. In addition, when playing a game, players enter into a "magic circle", or the space within which a game takes place [Huizinga, 1942], by adopting a lusory attitude. However, those prerequisites do not necessarily make play meaningful. According to Salen and Zimmerman, interactivity and decision are what made game play really meaningful. In the base of obeying rules, within the boundary of magic circle which builds up the psychological and emotional atmosphere, decision making enticingly leads the player to pursue the goal of the game [Järvinen, 2004].

There are fundamental differences between play and other activities. Play is a spontaneous activity, in which a player engages with the playing content that does not affect to other entities outside the play. Throughout the game playing process, the player is free to make any arbitrary choices by following the rules of play voluntarily, pursues for the victory driven by the external motivation of play, and finishes the play without any serious influence on outside the game [Malone, 1982]. On the contrary, other activities, taking an example of work, are existed for clear goals. An employee engages with the work tasks driven by the instinct of exchanging his or her effort to money or fame. The goal of work is to finish one's duty predefined in the work contract, anything deviates from achieving this goal may have potential influences on the employee and the employer. Playing games is a unique activity comparing to other non-game events, which can be differentiated in a manner of motivation, and engagement.

2.2. Motivation

The studies of motivation and how it influences human's different daily activities have been carried out throughout the years. This section introduces some of the most remarkable theories that influence game and play mindset. Subsection 2.2.1 presents the self-determination theory, subsection 2.2.2 introduces the cognitive evaluation theory, and subsection 2.2.3 explains the relationship between rewards and motivation discovered by researchers.

2.2.1 Self-Determination Theory

Deci and Ryan [1985] have introduced the Self-determination theory (SDT), which presents a motivational spectrum from the boundaries of amotivation, then different levels of external

motivation, to intrinsic motivation.

As shown in the Figure 1, the detailed motivation breakdown and their respective regulatory styles of the motivation spectrum is presented, along with the perceived locus of causality and relevant regulatory processes of every regulatory style.

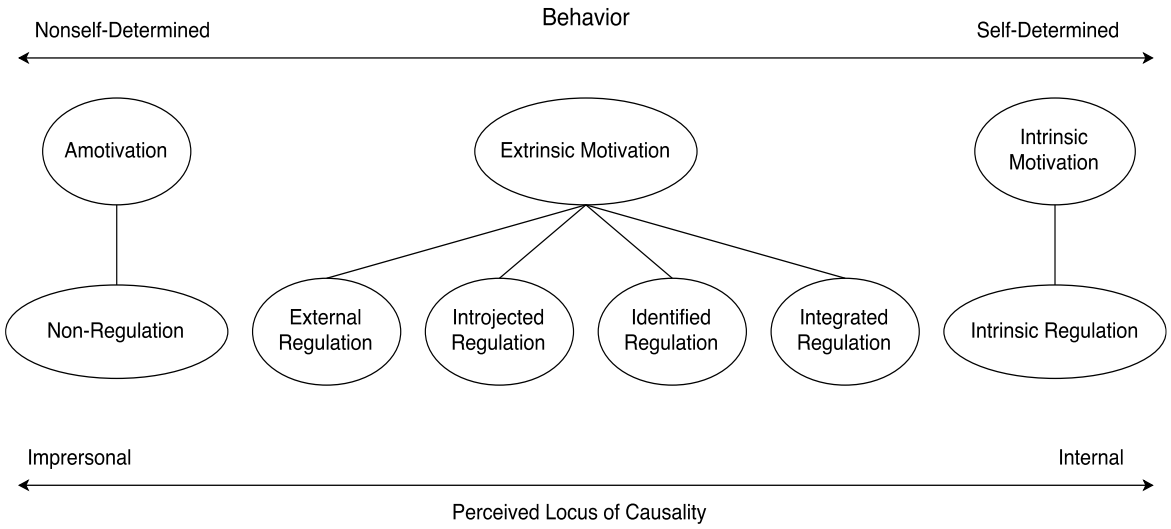

Figure 1. Motivation spectrum introduced by Deci and Ryan [1985].

Amotivation defines the complete apathy of performing any action, located in the extreme side of the motivational spectrum, it is totally not regulated and not influenced by any parties.

Extrinsic motivation consists of four extents in the motivation spectrum. The first extent which is nearest to amotivation is *external regulation*, where external orders or external needs are the only reason to do something. Next extent is *introjected regulation*, where a person replicates some behaviors, attributes or other fragments from the external atmosphere, in order to receive his or her own external motivation, for instance getting status or obtaining people's fondness by performing these behaviors. Next extent is *identified regulatio*n, which reflects a conscious knowledge of behaviors, such that the action is driven by his or her behavioral goal and hence of personal importance. *Integrated regulation* is the last extent of extrinsic motivation, which only occurs if identified regulation is entirely assimilated to one's own. Actions driven by motivation that considered to be integrated regulation share various qualities with intrinsic motivation, except that these actions are focused more on attaining separable outcomes instead of inner satisfaction.

Intrinsic motivation, located in the other end of the spectrum, is the natural virtue of preference towards spontaneous interest, exploration, mastery, and assimilation [Csikszentmihalyi and Rathunde, 1993], the intrinsic motivation is crucial for human cognition

and societal development and hence characterizes a primary source of pleasure, vitality and liveliness in life [Ryan and Deci, 2000].

2.2.2 Cognitive Evaluation Theory

Cognitive evaluation theory (CET) is a sub-theory for SDT presented by Deci and Ryan [1985] which aims for specifying factors that causes the variability in intrinsic motivation. There are three main characteristics in Intrinsic motivation which appear to be fundamental for promoting ideal performance of the natural tendencies for progression and integration, together with individual well-being and constructive societal development [Ryan and Deci, 2000]. The three characteristics are *competence*, *autonomy* and *relatedness*.

- *Need of Competence:* the desire to control and master the environment and outcome. Humans have the tendency to understand the process of things, and know the results of their actions.
- *Need of Autonomy:* the urge of act in harmony with the integrated self. Distinguished from the need of independence, humans favor to have sense of free wills when they are doing things and taking actions out of their own interests and values.
- *Need of Relatedness:* the desire to interact with, be connected to, and engage with caring for other people. Humans acquire the sensation of belongingness from making actions and doing daily activities involved with other people.

Deci and Ryan [1985] argues in the CET that intrinsic motivation for some actions can be enhanced by the feelings of competence conduced from social-related events, such as feedback, rewards, and communications. Accordingly, optimal challenges, positive performance feedback, and refrain from evaluations are all proved to improve the variability of intrinsic motivation. However, the feelings of competence will not enhance intrinsic motivation, unless escorted by a sense of autonomy.

Thus, in order to increase the effect of intrinsic motivation, it is not sufficient to only experience the efficacy or competence, the direct contextual support for autonomy is also of prior.

Although supports for the characteristics of competence and autonomy have outstanding influence on constructing variability in intrinsic motivation, the third characteristic relatedness, is tied with intrinsic motivation due to the human nature. Human has established the connection with others from the time being as an embryo. Studies [Frodi et al., 1985] have shown that both security and maternal support present more exploratory behavior for infants which can be interpreted as an intrinsic motivation. It is also presented by Ryan and Grolnick [1986] that a

lower level of intrinsic motivation is observed for students whose teachers are insensible and cold. A secure and warm connection appear to be essential for an individual to increase his or her intrinsic motivation.

2.2.3 Rewards and motivation

The relationship between rewards and motivation and how do rewards affect motivation have been analyzed by many researchers.

In the cognitive evaluation theory presented by Deci [1975], depending on the different classification method, rewards have two conflicting effects: *informational* and *controlling*.

Informational rewards are rewards which consist of positive feedback of one's performance competency. For example, a compliment for job well done, or a certificate proving the success of some activities. Because of the need of Competence, information rewards usually enhance intrinsic motivation. Rewards which consist of more *controlling* effect, on the side hand, will lead to the loss of intrinsic motivation, due to the fact that being controlled undermines the need of autonomy.

Rewards can be defined into many different categories based on different classifications. Some of the classifications are presented below.

- Rewards classified by tangibility:
  - *Intangible rewards* are rewards that don't have an inherent monetary value, and are usually awarded for a specific accomplishment. Verbal rewards such as thanks and compliment from others, and virtual rewards such as achievement badges are all identified as intangible rewards.
  - *Tangible rewards* are rewards that are tangible and have financial values, for example a lunch coupon, a phone, bonus or salary increase.
- Rewards classified by expectancy:
  - *Expected rewards* are rewards which their arrival and the content of rewards can be foreseen, or they are given in a regular routine.
  - *Unexpected rewards* are rewards that usually surprise receivers.
- Rewards classified by contingency*:*
  - *Non-contingent rewards* are rewards that are given no matter what.
  - *Engagement-contingent rewards* are given if the task is started.

- *Completion-contingent rewards* are given if the task is finished.
- *Performance-contingent rewards* are given if the task in perform well.

Deci et al. [1999] have presented the effects of extrinsic rewards on intrinsic motivation by meta-analysis.

The study indicates that all extrinsic rewards, including all tangible rewards, and all expected rewards engagement-contingent, completion-contingent, and performance-contingent rewards drastically undermine self-reported interest and free-choice intrinsic motivation.

On the contrary, intangible rewards, such as positive feedback, improve both free-choice and self-reported behaviors which will lead to enhancement of intrinsic motivation.

2.3 Engagement and flow

The definition and functionalities of engagement have been researched in various fields throughout the years, this section first elaborates early studies of engagement in subsection 2.3.1, then presents user engagement theory in subsection 2.3.2, and last explains the positive psychology (flow) in subsection 2.3.3.

2.3.1 Early studies of engagement

A notable definition of personal engagement is presented by Kahn [1990], which describes personal engagement as:

> "the harnessing of organization members' selves to their work roles."

Even though the research of Kahn is focus on the psychological condition of personal engagement in work, it has presented the first constructive definition of engagement and provided profound research background user engagement and disengagement studies.

Kahn also states that in work environment, many different work roles are divided and allocated. People not only have different roles in different working placement, but they also invest their personalities to these various roles. Network of communication roles are caused by these role allocations and divisions. The more people employ and express themselves physically, cognitively, and emotionally during role performances, the more personal engagement is involved. On the contrary, the uncoupling of selves from work roles reflects personal disengagement, which will lead to physical, cognitive, or emotional withdrawal and defense in role performances.

The personal engagement and disengagement concepts integrate that self-expression and self-employment are essential in work environment in order to increase people's self-cognition level, integration, productivity, and ultimately increase their motivation in work.

2.3.2 User engagement

The definition of user engagement is presented by Attfield et al. [2011], that user engagement is the emotional, cognitive and behavioral connection that exists, at any point in time and possibly over time, between a user and a resource.

Numerous characteristics of can be linked to user engagement. As presented by, based on three broad dimensions *emotional, cognitive,* and *behavioral,* eight kinds of characteristics are associated with user engagement. Table 1 presents the eight characteristics identified by Attfield et al., and their definition and measurement respectively.

Characteristic	Definition	Measures
Focused Attention	Focusing attention to exclusion of other things	Distorted perception of time, following on task performance, eye tracking
Positive Affect	Emotions experienced during iteration	Physiological sensors (e.g. face detection)
Aesthetics	Sensory and visual appeal of an interface	Online activity, Physiological sensors, perceived utility
Endurability	Likelihood of remembering an experience and the willingness to repeat or recommend it	Online activity (e.g. bookmarking, sending emails)
Novelty	Novel, surprising, unfamiliar or unexpected experiences	Physiological sensors (e.g. blood pressure)
Richness and control	Levels of richness and control	Online activity (e.g., interaction with the site, time spent), Physiological sensors (e.g. mouse pressure)
Reputation, trust and expectation	Global trust users have on a given entity	Online activity (returning user, recommendation)
User Context	User's motivation, incentives, and benefits	Online activity (location, time, history)

Table 1. the identified characteristics of user engagement introduced by Attfield et al. [2011].

User engagement is an important factor to consider when designing user-centered web applications, Thus, understanding the ideology of these characteristics is essential for designing engaging experience, which will eventually lead to the design of successful web applications that can keep users for persistent engagement.

2.3.3 Flow

The concept of flow in positive psychology is proposed by Csikszentmihalyi in the year of 1975. Flow is the mental state which enables people to execute an activity with optimal devotion, full involvement, and absolute enjoyment.

There are many conditions for people to reach the state of flow. As it is illustrated in Figure 2, to make flow occur, the activity should have a moderate difficulty; otherwise, the flow will not stay, as too complicated activity would easily lead to anxiety, and too easy activity would easily lead to boredom state.

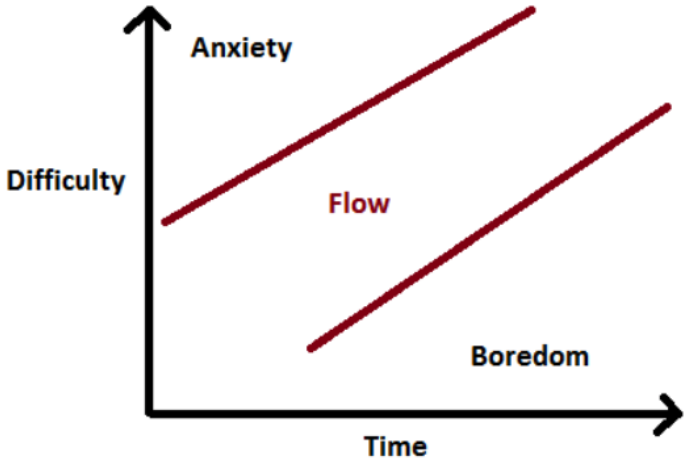

Figure 2. Optimal state for flow to occur [Csikszentmihalyi, 1975].

Additionally, clear goals, awareness and attention, balance between perceived challenges and perceived skills, immediate feedback are all linked to the necessity of flow [Csikszentmihalyi, 1975].

Maintaining a stable flow when creating a gamification system will not only promote user engagement in the system, but also reflect cognitive ability which enhances the user's self-confidence and self-integration.

3. Gamification

This chapter presents the ideology of gamification. The first section presents the history of gamification, different definitions of this concept by scholars. The second section introduces the effectiveness review of gamification. Section 3.3 defines the process of gamifying a system. Section 3.4 introduces several examples of different systems integrated with gamification. Section 3.5 describes the criticism and potential risks in gamification, and last section presents studies of use of gamification in fitness systems, as well as most applied game-like elements in fitness services.

3.1 Definition of gamification

The term gamification was first coined in 2002 by Nick Pelling, but it had never gained its popularity until the year of 2010 [Marczewski, 2012]. There are many definitions for gamification among scholars, one of the most cited definitions is "a design metaphor to use game design elements in non-game context" [Deterding et al., 2011].

However, according to the research of Huotari and Hamari [2012], there is limitation in Deterding's definition, owning to a lot practices of gamification can be happened in game-related context. Thus, a refined definition has been proposed: "a process of enhancing a service with affordances for gameful experiences in order to support user's overall value creation" [Huotari and Hamari, 2012]. This definition eliminates the restraint that gamification can only occur when game-like elements are applied in non-gaming contexts.

Another similar explanation has been presented by Marczewski [2012], who defines gamification as "The application of gaming metaphors to real life tasks to influence behavior, improve motivation and enhance engagement." In order to achieve the goal of increasing user motivation and engagement, a lot of practice of gamification have been applied in the social objects and business fields, such as marketing, education, work, health and fitness.

Because the term gamification is relatively new in academic society, and many scholars define the concepts related to gamification based on their liking. Thus, there are many similar or identical concepts which named differently. For example, *game elements, game design elements, gamification components, motivational affordances*, and *game-like elements* are all referred to the same idea by different scholars.

This thesis employs the definition of gamification given by Huotari and Hamari [2012]. In addition, unless it is directly quoting the scholar's definition, the thesis uses *game-like elements* to formulate the game elements utilized in non-game context.

3.2 Review of gamification

Since the concept of gamification has gained a lot of popularity both in academic world and business industry, immense number of researches have attempted to analyze the results of gamification.

According to the literature review of empirical studies on gamification which analyzes 809 peer-reviewed papers that consist of search terms of *gamification*, *gamif**, *gameful* or *motivational affordance* [Hamari et al., 2014], gamification does produce positive effects and benefits evaluated from most of the reviewed papers.

One of the negative results from the reviewed papers is, for example, the use of gamification might not be as effective in utilitarian services. However, it is unclear that the ineffectiveness is caused by lacking motivations of users or the nature of the gamified system examined by the papers.

Overall, the use of gamification can lead to promising outcomes, especially if it is applied to services related to education or learning, health or exercise, work, intra-organization, and innovation or ideation [Hamari et al., 2014].

3.3 Gamify a system

In order to gamify a system, its business problems should be primarily checked and the suitability of gamification for the system should be validated, the first subsection presents this examination. The second subsection providers the design guideline for gamifying a system. The last subsection introduces different game-like elements which are frequently integrated to gamification services.

3.3.1 Identify business problems

Based on the previously introduced review by Hamari et al. [2014], gamification does not fit for every situation.

Gamification is about activating the user's intrinsic motivation, and improving the user engagement in a system which originally lacks motivation. Therefore, it is important to identify the problems or requirements before utilizing gamification to a system. There are four questions to examine if gamification is right for the business problems [Werbach, 2017]:

1. *Motivation*. There are two situations where gamification is a right tool to deal with lacking motivation. One is when the activities are complex and unfamiliar which involve profound creativity, unique skills, or connections. The other is when the

activities are relatively uninteresting.

2. *Meaningful choice*. The activities or designed choices are sufficiently attractive for user care about the outcome.
3. *Structure* of the activities and their behaviors can be encoded in rules and algorithms.
4. *Potential conflicts* with other motivational structures. For example, gamification in schools may decrease the intrinsic desire of learning of students, or gamified systems in work environment may be neglected because of the tension of employees to keep the jobs.

3.3.2 Follow the incentives

Once the business problems have been identified to be suitable of gamifying, it is possible to start the design of gamification system. According to the gamification design guideline by Werbach and Hunter [2012], two major incentives for performing activities are presented.

- Incentive for *collective goods*. For activities of which a majority of users consider as uninteresting or complicated, the use of PBL system (points, badges, and leaderboards presented by Hamari and Eranti [2011]) can be beneficial. The playfulness of collective goods can lead to a coherent and consistent flow, which will ultimately improve the engagement of users and promote their intrinsic motivation. For example, in a system for studying a foreign language, the use of gamification not only helps learners in numerous personality factors, but also enables learners to transfer from shy and introvert to more positive and motivated modes [Flores, 2015].
- Incentive for *happiness*. For activities that involve a lot of engagement and usually hard to persist, following the criteria from creating a flow by Csikszentmihalyi [1975], e.g. providing clear goals, appropriate challenges, and immediate feedbacks, is constructive for promoting happiness and fulfillment to users, which will also encourage users to voluntarily stick with the systems.

In many cases, systems to be gamified usually contain activities executed by both incentives. Thus, combining these two approaches in building gamified systems is conceivable and commonly practiced in the market.

3.3.3 Game-like elements

The review of gamification researched by Hamari et al. [2014] has presented that there are ten different categories of game-like element (including motivational affordances) that being used

in most of the gamified systems. These ten game-like elements are: *Points, badges or achievements, leaderboards, levels, story or theme, clear goals, feedbacks, rewards, challenges, and progress*. Among them, the first three elements (also known as the PBL system) are the most commonly found variants in systems.

- *Points* are figures in numeric forms based on data in some specific fields, such as hours spent on studying, and number of posts being liked. The element of points is the foundation of other game-like elements. For example, points can be used for keeping track of scores, determining win states, connecting to rewards, providing feedback, and displaying progress.
- *Badges* or *achievements* are representations of collectable virtual goods. They not only enable users to monitor and organize their actions, but also increase awareness of users of their skills rank among peers [Jakobsson and Sotamaa, 2011]. In gamified systems, it is common that a badge or an achievement is automatically acquired once the requirement is fulfilled. For example, a "three-days-in-a-row" badge is given if the user has logged into the system three days in a row. This game-like element works because the human's natural incentive for happiness can be earned by collecting limited goods and the completion of collection.
- *Leaderboards* are ranking components which can express the feedback on competitions by displaying the same category of points between different users. Leaderboards can be set in a small scope, where only limited users are chosen for the comparison (e.g. leaderboards within friends). Leaderboards may demotivate users if they see one person is too far ahead and there is no way to catch up [Werbach, 2017]. However, good design of leaderboards (for example, scores in leaderboard reset every week, or some special aids to the users in the board) can dismiss this problem.
- *Levels* are usually representations for the progress of users. A higher level indicates the superior status of a user and may unlock more collectables. There are two kinds of leveling systems: infinite level system and finite level system. In an infinite level system, the effort of incrementing each level is the linear and there is no cap to continue leveling. This system may cause troubles for demotivating newer users, because they can never catch up. Hence, in order to balance this situation, a finite level system is generally accepted and utilized, where leveling is exponential and reaching the maximum level is very difficult but will give a great status and fulfillment for users.
- *Story* or *theme* can provide sparkling and imaginary background for a system. *Story*

offers a narrative thread to user in the whole lifespan of the system. There are four characteristics when designing a story for a system: *characters* (they can be either the users themselves, or some mythological figures), *plot* (the structure of the whole story), *tension* (potential conflicts to move the story forward), and *resolution* (the results lead by actions and tensions). On the other hand, *theme* is a lighter version of *story*, which usually contains background story, rules, and supporting aesthetics design [Boller, 2013].

- *Clear goals*, for examples, to obtain all achievements, or to reach the highest level, are the ultimate achievements by using the system. On the way of a user aiming for the goals, the principles of increasing user engagement and improving motivation of the system is fulfilled [Deterding et al., 2011].
- *Feedbacks* are aimed to facilitate and support users [Huotari and Hamari, 2012]. For example, in a gamified language learning system, a feedback is given when a user finishes the learning task for the day. Even though feedbacks are utilized in all kinds of fields, the *feedbacks* in gamification refer to the interaction and support provided by the gamified system.
- *Rewards* are dispensed as in-system goods in order to motivate users to level up, use the system more, or reach for their goals. Three types of rewards are categorized by Duggan and Shoup [2013]: *Recognition* (e.g. reputation and status conferred and displayed to other users), *privileges* (e.g. early or special access, moderation ability, or stronger votes), *monetary rewards* (e.g. discounts, free delivery, prizes, and redemptions).
- C*hallenges* (e.g. finish a task within a time constraint, be top three in the leaderboard) are calls to prove or justify one's ability or strength. Due to human's competitive nature, challenges are motivational for users to do tasks that they do not usually do [Epstein, 1980]. However, this element should not be abused because there are huge differences in people's competitive levels, too heavy challenges may lead to causal users abandoning the system.
- *Progress* can act as the completion indicator of goals, the accomplished achievement indicator, and also the stimulation of challenges.

The psychological and behavioral outcomes by these ten game-like elements have been proven positive by Hamari et al. [2014]. Henceforth, based on the purposes and requirements of designated system, the process of gamifying can be achieved by selecting suitable game-like elements and integrating them into the system.

3.4 Examples of gamified system

In this section, three gamified systems classified in different categories are presented.

3.4.1 Reddit

Reddit is a web service which contains social news aggregation, web content rating, and discussion [Reddit, 2017]. Divided by topics, variously subsections build up the whole system, which are called subreddits.

Unlike other social news systems, gamification has been well integrated into the fundamental structure in Reddit, which has helped gaining a lot of popularities, especially among young users. Owning to the attractive features provided by gamification, users in Reddit share individual experiences, support educations, give advices, or even send presents, even though most of them have never met each other [Richterich, 2014].

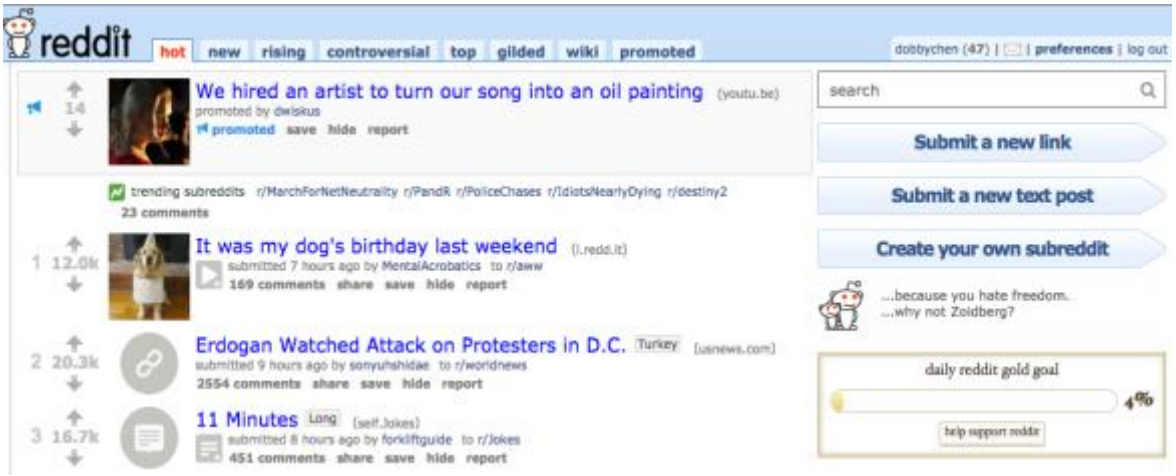

Figure 3. Reddit website (https://www.reddit.com).

As it is shown in Figure 3, many game-like elements can be found in the front page of Reddit. For example, element of *points*, namely *karma-points* in Reddit, is the most essential gamified feature in Reddit. Each user can influence the *karma-points* of posts by upvoting or downvoting the posts contributed by other users, depending on the quality of the posts or, their personal preferences. The posts which have the most *karma-points* are shown in the top view of the system, which indicates direct, fair and numerical representation of post assessments [Richterich, 2014].

Other game-like elements, such as clear *goals*, *levels*, and *challenges* are also found in this famous gamified system.

3.4.2 WordDive

Education is one of biggest targeting field for gamification, because the use of game-like elements can improve user's engagement and provide positive motivational effects [Hamari et al., 2014].

WordDive is an e-learning service where users can study foreign languages online, in a gamification way. WordDive works without any additional paper-form material, the learning process only requires language learners to have their electronic devices, such as mobile phones, or computers. In order for users to learn a new vocabulary, the application provides a picture of hint, a description of the word, and the synonym if available (see Figure 4). During the exploration and repetition of this process, the users are able to remember many vocabularies without losing interest [Scheid, 2015].

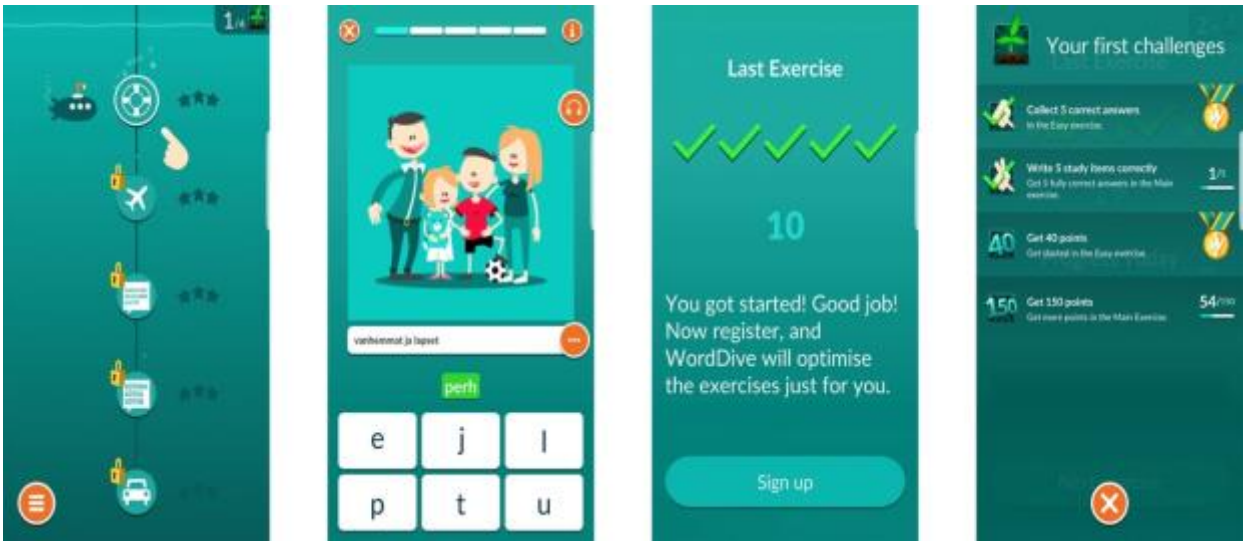

Figure 4. Worddive mobile application (android).

Many game-like elements, such as clear *goals*, *progress*, *points*, *challenges*, and *theme* are integrated in WordDive, making it one of the most popular language e-learning services. As of May 2017, WordDive has over three hundred thousand users in 150 countries [Worddive, 2017].

3.4.3 OASIS

Although most of the gamification practices are related to software services, gamification does not necessarily to be only integrated to software services. As long as a service is enhanced by gameful experiences and the user's overall value creation is supported, the act of gamification is fulfilled [Huotari and Hamari, 2012].

For example, OASIS is an open, social, and playful environment located in the University of Tampere. OASIS encourages open culture, informal learning and casual information sharing [Kultima et al. 2015]. Figure 5 shows the interior design of this playful, elevated auditorium-like space.

In this space, many services are provided and hence, gamification has been integrated into these services. For example, the element of *theme* is integrated to the whole space, and the elements of *rewards* and *feedbacks* are utilized in various seasonal experiments.

Figure 5. OASIS[1]: a playful environment in the University of Tampere.

In the experiment: *OASIS Deck of Cards* which focuses on community building, many gamification approaches for this experiment is discussed, to help building the motivation for playing [Nummenmaa et al., 2015].

As a famous and beloved environment by students and staffs in the University of Tampere,

[1] Picture accessed from Oasis official website: https://oasis.uta.fi

there are a lot more research experiments to perform in the future, where gamification would have a key role in these experiments.

3.5 Criticism and risks of gamification

The practice of gamification is to learn from gameful experiences and apply values of game in other contexts. Despite the fact that many researchers have presented positive reviews on gamification, there are some who argue that it does not accomplish the value creation but only does harm. According to Werbach [2017], there are four main categories of criticism and risks. Subsection 3.5.1 presents a view of pointsification, subsection 3.5.2 presents a term called exploitationware, subsection 3.5.3 talks about abuses of gaming mindset, and finally subsection 3.5.4 and subsection 3.5.5 describe the potential legal and regulatory issues.

3.5.1 Pointsification

The term pointsification was coined by Margaret Robertson [2010], who argues that people tend to integrate the PBL elements (points, badges and leaderboards) to a system and call the process gamification, which are not the essential merits in experiences of games. For the process of implementing systems which claimed to have been gamified but only integrated PBL system, Robertson insists the process should be named pointsification, because it does not reflect the powerful stimulation of motivation merits in games and will cause ineffectiveness.

It is true that comparing to other game-like elements, PBL system is applied at a highest number [Hamari and Eranti, 2011]. Thus, to achieve gamification, one should not just integrate PBL system without other consideration. As it is described in section 3.3, one should do research based on the requirement and background before gamifying the system, and avoid the thoughtless pointsification.

3.5.2 Exploitationware

On the contrary of what Robertson [2010] has suggested (the abuse of pointsification which makes gamification ineffective), the term exploitationware is coined by Bogost [2011] for the potential of gamification being too effective.

As a reason of tremendous effectiveness, Bogost argues gamification can be used for getting people to do things which are not necessarily in their interests. Particularly in the work environment, the abuse of gamification can fundamentally undermine the nature of economic and social exchange, as gamification proposes to replace real incentives with fictional ones [2011]. For example, in the laundry rooms of Disneyland and Paradise Pier hotels in Anaheim,

the use of gamification (a leaderboard was displayed in the giant monitor which showed each employee's working speed) was intended to keep productivity and increase motivation by competition [Lopez, 2011]. This act was later abolished, because it was not encouraging and motivating the workers, but instead controlling and manipulating them to compete more aggressively to keep up [Werbach, 2017].

The unethical abuse of gamification or exploitationware should be avoided, especially when gamification is developed in work related systems.

3.5.3 Gaming mindset abuses

The targeted users of gamification systems are humans. Even though the behaviors and motivational structures can be anticipated when designing a system, sometimes it is hard to predict what users really do [Werbach, 2017].

One of the most common abuses by gaming mindset is cheating. Because it is easy for users to feel a gamified system as a game, sometimes they don't take the consequence of breaking rules so seriously. If any design flaws of the gamified systems are found, some would make use of these flaws and cheat to get the incentives or social rewards. Another unintentional abuse is reported by Lazzaro [2012], which is focusing on the gamification tasks over the human system of engagement for personal profits. Lazzaro states that the excessive focus of personal profits can decrease the meaning of the systems and also draw secondary effect on other users in the systems [2012].

In order to eliminate these abuses, a robust design of the gamified system is required. In addition, a good understanding of the targeting users, e.g. stimulate how they think, what they want from the system, is of great importance to protect the system from being ruined by abusers.

3.5.4 Legal issues

There are a few legal issues needed to take considerations when gamifying a system [Werbach, 2017].

Privacy is a big aspect of gamification legal issues, because in order to display all the game-like elements, a system acquires user's information and process it visually in the system. Thus, it is important for the system to protect all this private information from abusive operations.

Employment or *labor law* should also be contemplated when gamifying a work-related system, because in some areas the right of employ is restricted (game-like elements cannot be used for affecting people's work condition in some countries [Werbach, 2017]).

Deceptive marketing can occur when some advertisements are disguised as a form of gamification. It is acceptable for some systems which can be easily recognized as advertising systems; however, if a system misleads users that there would be some rewards for using the system which turns out to be a deceptive advertisement, this act is not only problematic and unconscionable, and in some countries, it can also violate the marketing law.

Intellectual property law regulates people's access to information or digital assets. That is to say, using some other people's digital property without crediting or paying can cause a big legal conflict. Especially when designing of rewards or achievements, one should be cautious to design something too similar to someone else's work. Otherwise, plagiarism of intellectual properties may be violated.

Virtual property rights can be offended, for example, if the system provider alters the content of virtual assets. However, in the majority of countries, there is no explicit law defining the owner of virtual properties. Thus, explaining the detail rules of virtual property in terms of service to avert future conflicts is advisable.

3.5.5 Regulatory issues

In addition to legal issues, there are several regulatory issues may be found in gamification:

Paid endorsements. There are some game-like elements in some gamified systems are only obtainable by endorsing the system. For example, users have to like the system's Facebook page to unlock an achievement, or to attain a chance of draw to get some tangible or intangible rewards. The idea behind is act for the system sides is to broaden the market by sharing. However, this can be abusive and deteriorate the gamification experience.

Banking regulation should be checked if there is financial or trade affiliated with the system, or there is tradable structure around a virtual currency.

Gambling on obtaining some special badges or rewards is allowed and under regulation in most countries. Even though slot machines are also gamified systems, one should acquire the approval by regulation before implementing a gambling-related system.

3.6 Gamification in fitness services

Since gamification is broadly used by researches and technology from the year of 2010, it has gained its popularities in the health and fitness field. Fitness and health related companies have widely accepted and adopted gamification as a means to increase initiation and retention of desired behaviors [Fankhauser, 2013].

In the year of 2013, whilst gamification of fitness and health was still in its infancy, Dominic King and his colleagues have noticed the integration of gamification was trending, but the number of issues lied in the integration which might affect the success of gamification were massive [King et al., 2013].

Focusing on this concern, Lister et al. [2014] conduct a research for analyzing gamification and health behavior in top downloaded applications from Apple App store. From the results of identification and measurement of a total number of 132 sample applications, the descriptive statistics have been categorized into three main measurement rubrics: *behavioral constructs*, *game elements*, and *gamification components* [Lister et al., 2014].

Behavioral constructs are based on the behavior theories. Table 2 presents three different behavioral components and their corresponding details. Modern mobile technology has the *capacity*, *motivation* and *opportunity* to offer personal health-related and fitness data and provide timely behavioral prompts [Wu et al., 2012]. Self-monitoring, self-efficacy, and goal-setting are the highest constructs among all behaviors.

Capacity	General information	Self-monitoring	Stress management	Skills training	Simplicity or enabling factors
Motivation	Incentivization	Social support (positive reinforcement)	Goal-setting	Cognitive strategies	Self-efficacy
Opportunity or **trigger**	Peer pressure	Cues to action	Stimulus control		

Table 2. Three types of *behavioral constructs* [Lister et al., 2014].

There are thirteen different *game elements* (illustrated in Table 3) identified in the fitness and health applications reviewed and measured by Lister et al [2014], among which, after-game feedback or reinforcement, self-representation with avatars, and parallel communication systems are the top three game elements that are integrated to the systems.

Self-representation with avatars	3D environments	Narrative context	Feedback before or during game	After game Feedback or reinforcement
Leaderboards	Ranks of achievements	Different levels of play	Marketplaces and economies	Competition under rules explicit and enforced
Teams (multi-player modes)	Parallel communication systems	Time pressure		

Table 3. Thirteen *game elements* [Lister et al., 2014].

Six g*amification components*, as Lister et al. [2014] describe, are identified from the reviewed applications. Table 4 presents the details of these *components*. Social or peer pressure, digital rewards, competitions or challenges are the top three components applied.

Leaderboard	Levels of achievement or rank	Digital rewards
Real world prizes	Competition or challenges	Social or peer pressure

Table 4. Six *gamification components* [Lister et al., 2014].

The evaluation results from Lister et al. [2014] show that the use of gamification components, game elements, and *behavioral constructs* are overall abundant in health and fitness applications, where *gamification components* and *game elements* have positive impacts on targeting motivations and triggering user engagements and rising popularity of applications, and behavioral constructs have potentials for changing user fitness or health related behaviors and hence improving outcomes.

According to the Lister et al. [2014], even though burgeoning use of *gamification components* and *game elements* have been identified in health and fitness applications, standard guidelines for integration these elements has been neglected by the industry.

As it is introduced in section 3.1, there are various definitions for the term gamification and its related concepts. The research outcomes from Lister et al. are promising; however, the classification of *gamification components, behavioral constructs,* and *game elements* is vague and confusing. For example, *Leaderboard* is defined in both *gamification components* and *game elements,* and *peer pressure* is included in both *gamification components* and *behavioral constructs.* Based on Huotari and Hamari's definition [2012], gamification does not only include utilizing game elements in non-game context, the gameful experiences which can enhance services should be also included as game-like elements. Therefore, combining the most used elements identified by Lister et al. [2014] with the game-like elements in general gamification system introduced in subsection 3.3.3 [Hamari et al., 2014], this thesis proposes a reformative list of game-like elements for Fitness systems: *Points, badges or achievements, leaderboards, levels, story or theme, clear goals, feedbacks, rewards, challenges, progress, avatars, parallel communication,* and *peer pressure.*

Even though game-like elements are not whole of the gameful experiences, the main focus of integrating gamification to fitness service is undeniably the integration of game-like elements [Pereira et al. 2014]. That is to say, in the scope of this thesis, selection and implementation of game-like elements are the core process for integrating gamification to web fitness services.

4. Web services architecture

This thesis focuses on integrating gamification to modern web services; thus, understanding the background of web services is also essential. The iteration of web service development has been evolving fast since the invention the World Wide Web [Berners-Lee, 1989]; hence, section 4.1 briefly introduces the definition and history of web service section 4.2 introduces the latest prominent architectures of web services that are used, section 4.3 presents the REST API which provides an application programming interface for either web servers or web clients, section 4.4 discusses the securing methodology for REST APIs, and lastly section 4.5 elaborates extensive concepts of web services.

4.1 Definition and history

Even though the World Wide Web was invented in as early as 1989, the limited speed and narrow coverage had made the web extremely rare and hard to spread in the early age. However, with the bursting dot-com development in 2001, the web has explosively risen and numerous of web services have appeared and utilized for consumers [O'Reilly, 2005]. To keep up with the expanding technology and set up a unified definition, the w3c organization has standardized the definition of web service as "a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks" [Booth et al., 2004], which is also considered to be the starting of era of web 2.0 [O'Reilly, 2005].

HyperText Markup Language (HTML) is the most basic building block of web services since the beginning of the web [Graham, 1995]. In web 1.0, hypertexts were the core component, as the hardware and technology condition had limited the transfer of content. However, due to the synchronous nature of architecture of web 1.0 which performs actions synchronously, users were gradually dissatisfied with the waiting time of web services. Hence, with the introduction of AJAX (Asynchronous JavaScript and XML) in the web clients, web services in web 2.0 have the ability to load displaying data without refreshing webpages [Gassner, 2013]. Comparing to older web services, O'Reilly also states that web 2.0 is capable of handling the machine communication process and delivery of multiple formats of data, such as pictures, videos and graphical texts [O'Reilly, 2005].

4.2 Modern web architecture

The most popular technology of web service used in the modern era is the service-oriented architecture (SOA) [Barry, 2017].

The service-oriented architecture essentially consists of a group of services which communicate with between. Typically, the communication is taking part between a web server and a web client over a HyperText Transport Protocol [Kalin, 2013].

4.2.1 Web Server

Web server, also as known as the web backend server, is the data access layer of the web service [Zimmermann et al., 2004].

All essential components and data associated with web content, along with the algorithm related to data computing and storing are hosted in the web server. For example, web server is responsible for reading and writing data from database, validating HTTP requests from the client side and responding respective information upon requests, if these requests are computed as secure and valid.

Currently there are numerous technologies for implementing backend servers, some popular server-side programming languages and their server-side targeting frameworks [Code, 2017] are briefly introduced:

C#: developed by Microsoft, C# and its web server framework ASP.NET[2] are typically applied by businesses which involve large database management. It is the second most used programming language for server-side deployment [Web, 2017].

Go: created by Google, it is a programming language which focuses on performance optimization. Gorilla[3] is a web toolkit for Go programming language which includes backend development support.

PHP: Designed solely for web development, PHP has preponderantly dominated server-side programming market, taking over 80% of the share of entire field reported by Web Technology surveys [Web, 2017]. One of the most popular framework for PHP backend programming is Laravel[4].

Java: As one of the oldest and most broadly adopted programming languages, Java found its way for implementing web servers in the early 2000s as a format called JSP [Code, 2017]. Spring[5] is a Java based framework which aims for building simple, portable, fast and flexible web server-side applications.

---

[2] Microsoft ASP.NET framework: https://www.asp.net/mvc
[3] Gorilla for Go: http://www.gorillatoolkit.org/
[4] Laravel for PHP: https://laravel.com/
[5] Spring for Java: https://spring.io/

Ruby: Originally popular in Japan as touted elegant and productive programming language, Ruby has gained its popularity over the world since its Ruby on rails[6] was added as a web server-side framework.

Python: arguably known as one of the clearest and most elegant programming languages, pythons is widely used in high-level general-purpose programming environment. As one of pythons' web framework, Django[7] fulfills the demand of design and maintenance of complex applications [Django, 2017].

Node.js: Even though JavaScript was only used for web frontend developing language to manipulate with user interface, as its popularity gains, JavaScript has been formulated to be utilized as a server-side language over the years as Node.js. Some notable server-side frameworks for Node.js are Express.js[8] and Hapi.js[9] and Koa.js[10].

As technologies involve and develop, increasing alternatives of server-side programming languages are introduced and applied for web services. Each language has its advantages and shortcomings, it is up to backend developers' preferences and companies' high-level business plan to choose which language to implement the servers for their web services.

4.2.2 Web Client

On the contrary of backend server, the web client is also called web front end, which is functioning as the presentation layer for web services [Zimmermann et al., 2004]. The web client, typically a web browser, fetches all necessary information from the web server in order to display contents in a human-friendly way.

The main building blocks for frontend are HTML, CSS (Cascading Style sheets) and JavaScript:

HTML has been the core component of the user interface of web services since the beginning ear of web [Graham, 1995], which describes and defines the content of a webpage.

CSS is used for describing the presentation of a HTML document, such as decorating the font, beautifying the layout, and animating the component.

JavaScript in client-side is used for interacting with different parts of components in HTML

[6] Ruby on rails: http://rubyonrails.org/
[7] Python Django: https://www.djangoproject.com/
[8] Express for Node.js: https://expressjs.com/
[9] Hapi.js for Node.js: http://hapijs.com/
[10] Koa.js for Node.js: http://koajs.com/

documents, manipulating the behavior of HTML document object model, and often being used as the API to communicating with the web server.

In recent years, demands for robust functionalities in user interface and striking user interface have driven the great development of front end; hence, there are many JavaScript Frameworks with affluent supports developed for enhancing the front end of web services, for example:

JQuery[11] is a light-weighted, fast, and cross-browser supported JavaScript library which minimizes the effort for traversing and manipulating HTML document, handling web events, animating, and simplifying Ajax.

Angular[12] is a JavaScript framework developed by Google which enables frontend servers to be cross-platform developed, optimized in great speed and performance.

React[13] is a declarative and component-based JavaScript Framework developed by Facebook, beloved with its efficiency and flexibility for building web user interface.

Vue.js[14] is a progressive JavaScript Framework which is approachable and versatile, also optimized in great performance.

Similar to different technologies available in server-side programming, choosing which JavaScript framework to implement frontend services is totally based on personal preferences, as most of them are optimized performing and well documented.

4.2.3 Communication

In web services, the web server and client are architecturally independent from each other, which underlies the communication between these two ends to secure the integration of the web service. HTTP is the fundamental protocol for this communication.

In order for web clients to make requests to web servers, messages are sent from web clients using the format supported by HyperText Transfer Protocol; in return, web servers respond respective messages using predefined format using the HTTP, some of the most common used responding formats are XML, SOAP, and JSON.

XML: shorted for Extensible Markup Language, XML providers the first document encoding format that has readable for both human and machine [Bray et al., 2008].

---

[11] jQuery: https://jquery.com/
[12] Angular: https://angular.io/
[13] React: https://facebook.github.io/react/
[14] Vue.js: https://vuejs.org/

SOAP: shorted for Simple Object Access Protocol, SOAP is a particular XML format designed for securely exchanging structured information.

JSON: shorted for JavaScript Object Notation, JSON provides both light-weighted and readability format, which is generally welcome by modern web services developers.

As it is introduced, there are many ways of communication for using HTTP. Among these methods, Representation State Transfer (REST) architectural style stands out due to its *stateless*, *uniform interface*, *cacheable*, *layered*, *client to server*, and *code on demand constraints* merits [Fielding, 2000].

4.3 REST API

Application Programming Interface based on REST architecture has gradually become the modern trends for building API for web servers [Gassner, 2013]. Web API built in the server side is considered as the common space for performing actions and responding data to web client. For REST APIs, HTTP verbs are used for requesting actions, and specified formatted document and metadata are responded agented by the REST APIs from the web servers [Masse, 2011].

4.3.1 Requesting action using HTTP verbs and URIs

Due to the constraint of uniform interface in REST architecture, the HTTP verbs embrace a key share for providing the action counterpart to the noun-based resource. There are four primary HTTP verbs: POST, GET, PUT/PATCH, and DELETE, which correspond create, read, update and delete operations respectively in database management. When making a HTTP request, the minimum statement is the combination of a HTTP verb and a Uniform Resource Identifier (URI), whereas sometimes OPTIONS and HEAD methods are also included for providing additional information.

GET is the verb for reading a representation of a resource. For example, pressing a link in the web client will trigger the GET method which will display new webpage for user.

POST is the verb for usually creating new resources. In addition, the POST method is also utilized for user credential authentication, along with OPTIONS method.

PUT is the verb utilized for updating existing resources with a brand-new content.

PATCH is the verb also used for updating existing resources. Instead of requiring complete resource using PUT request, PATCH request only requires providing changes of the resource.

DELETE is used for deleting a resource identified by a URI.

4.3.2 Receiving the data

The REST API responds with a response metadata and corresponding document, either the document is in HTML, JSON, XML or SOAP format. The metadata indicates the status for REST API of handling the request. A three-digit status code and reason-phrase are always responded by the REST API, unless the connection is broken.

Status code	Indication	Example phrases and explanations
1xx	Informational	100 Continue: The client must continue with its request.
2xx	Success	201 Created: The request has been fulfilled and resulted in a new resource being created.
3xx	Redirection	304 Not modified: If the client has performed a conditional GET request and access is allowed, but the document has not been modified, the server SHOULD respond with this status code.
4xx	Client error	401 Unauthorized: The request requires user authentication.
5xx	Server error	501 Internal Server error: The server does not support the functionality required to fulfill the request.

Table 5. HTTP Status codes and their indications [Masse, 2011]

4.3.3 Securing REST APIs

In most cases, especially when the service is related to private information handling, it is important to protect the privacy and security of protected resources. There are several ways to ensure the security of REST APIs.

For example, one way is to utilize API management solution vendors such as API reverse proxy, which offers reverse proxy-based services to address many cross-cutting concerns related to producing, and consuming, high-quality REST APIs. [Masse, 2011] Another way is to build the REST API with Open authorization framework, namely OAuth, OAuth 2.0, or OpenID Connect framework on top of the REST API. This thesis mainly introduces OAuth 2.0 for REST API securing method.

4.4 OAuth 2.0

There are four roles involved during an OAuth authorization flow, namely *Resource Owner*, *Resource Server*, *Client*, and *Authorization server* [Hardt, 2012].

Resource Owner is an entity which is able to grant access to a protected resource. For example, in a fitness service, the end user of the service is the resource owner.

Resource Server is the server which hosts the protected resource and handles the authorized requests by responding protected resource to the access token bearer.

Client is the application which requests for the protected resource within the authorization of the *resource owner*. The title of this section, *services* are presented as the *clients*.

Authorization Server is the server which issues access token to the *client*, given the completion of authentication of *resource owner* and authorization to the *client*.

In some circumstances, *resource server* and *authorization server* are hosted in the same server. However, it is possible that a sole *authorization server* issues access tokens to numerous *resource servers*.

4.4.1 The overview flow of OAuth 2.0 framework

In OAuth 2.0, in order for a resource owner to successfully fetch protected resources from a resource server by using a client, authorization and authentication must be undertaken prior to declaring a request; otherwise, the resource server will respond with an error message instead of the protected resource.

Firstly, a client sends the authorization request to the resource owner and waits for the corresponding authorization grant allocated by the resource owner. There are four possible ways of authorization grant, which will be described in subsection 4.4.2. Secondly, the client exchanges an access token for the previously gained authorization grant to the authorization server. Lastly, the protected resource can be accessed by the client providing the data request along with valid access token. Figure 6 presents the abstract protocol flow proposed by OAuth 2.0 framework [Hardt, 2012].

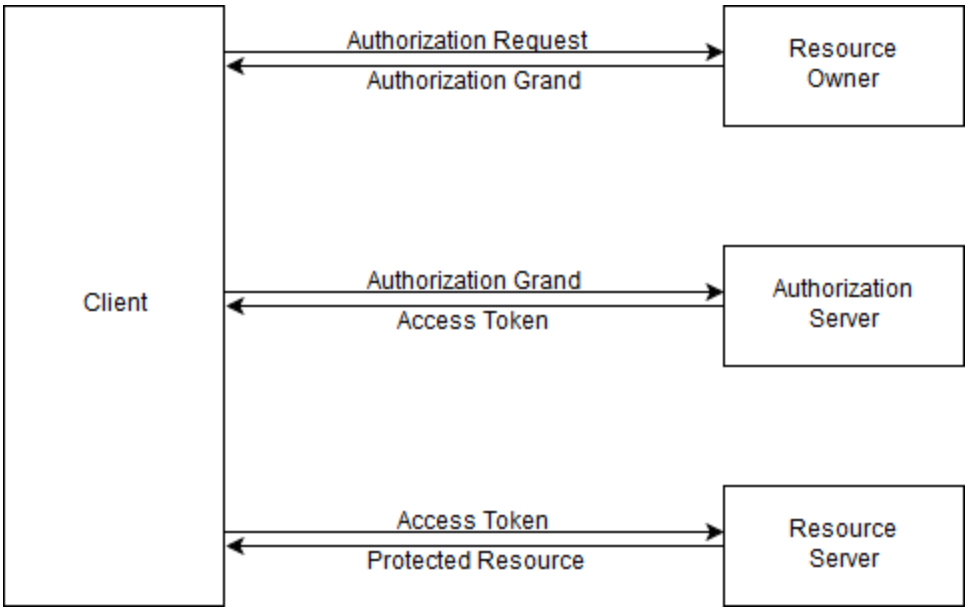

Figure 6: OAuth 2.0 Protocol Flow

4.4.2 Obtaining consent by Authorization Grant

There are four grant types provided by OAuth 2.0 Framework, which are *authorization code grant*, *implicit grant*, *resource owner password credentials grant*, and *client credentials grant*. It is up to resource servers to decide what authorization grant flows are provided for external clients to use.

Authorization code grant is a grant type which utilizes an authorization server as an intermediary between client and resource owner [Hardt, 2012]. In order to fetch an authorization code, the client redirects the resource owner to authorization page provided by the intermediary, which redirects the resource owner along with an authorization code after successful authorization. Subsequently, the client exchanges the authorization code for an access token and a refresh token from the intermediary, which can be stored in the client in order to proceed with resource requesting. Figure 7 illustrates the process of using OAuth 2.0 authorization code grant to access Fitbit REST API.

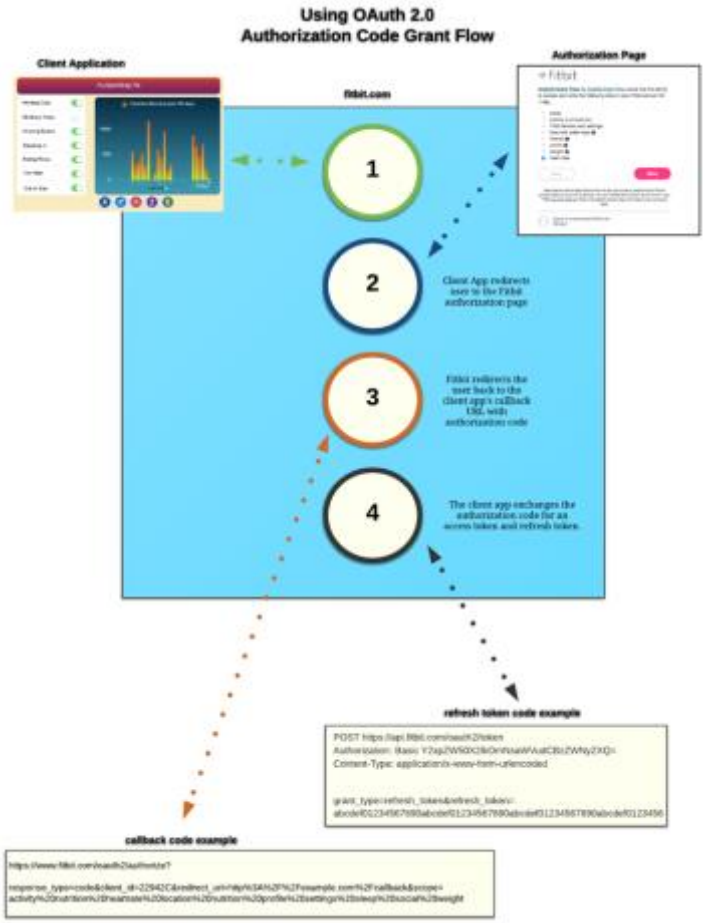

Figure 7. Authorization code grant flow provided by Fitbit OAuth 2.0.

The authorization code grant flow serves several particular security benefits, due to the fact that this grant flow requires server-to-server communication using a client secret attribute of a client, and the transmission of the access token is undertaken solely between the client and authorization server without the potential threat of exposure to other parties. Hence, it is recommended to web services which handle both the resource requesting and data storage [Fitbit, 2017a].

Implicit grant is a simplified authorization flow which enables clients that implemented entirely using scripting language such as JavaScript to securely request resource in the web browser of resource owner. No server-side code is required for implicit grand flow, instead of issuing an authorization code to the client, the authorization server issues an access token directly to the client along with the redirect URL registered in the resource server. Client secret key is not exchanged in the engagement of implicit grant flow; hence, the authorization session may not be extended automatically as no refresh token is involved in this flow.

As the implicit grant flow is designed for a light-weighted and simple flow, there are some limitations for using only implicit grant to build a complex web service with data storage in its

own sever. However, if the requirements of the web services are just securely fetching resource owner's data provided from the resource servers, and present them in visual-friendly way or integrating gamification, the use of implicit grant flow for authorization is totally adequate. In addition, providing the additional authorization method of using implicit grant flow in a complex web services may improve the versatility of the services, as well as increase the responsiveness and efficiency of the clients as fewer round trips are needed to obtain access token in this flow [Hardt, 2012]. Figure 8 demonstrates the process of using OAuth 2.0 Implicit grant to access Fitbit REST API.

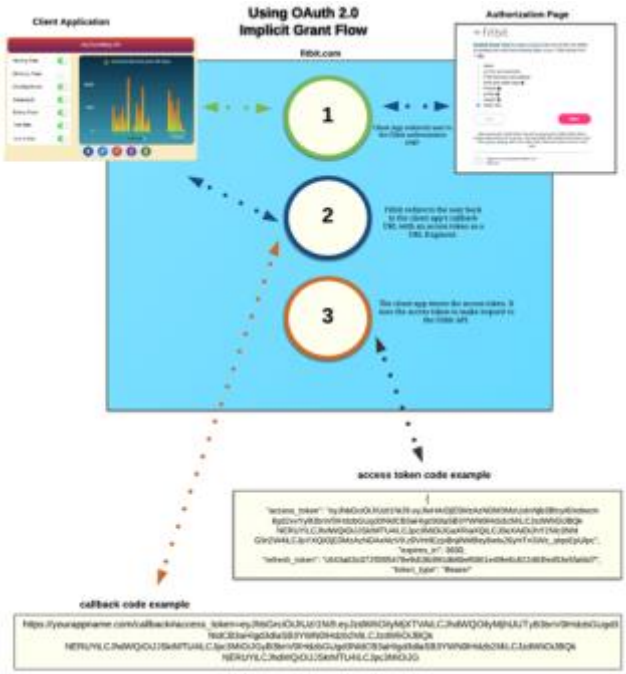

Figure 8. Authorization code grant flow provided by Fitbit OAuth 2.0.

Resource owner password credentials grant is a special authorization method which requires resource owner's credential information (such as username and password) to grant an access token and a refresh token. The beneficial reason of using this grant flow is that even though password credentials are accessed by the client during the authorization process, only a single request is required for the resource servers issuing the tokens. However, strong trusts between the resource owners and the clients are required for using this flow, as the password credentials are revealed in the clients, which may cause severe security potentials. Owning to this reason, resource owner password credentials grant flow is not supported by most resource servers, and it is not recommended for granting authorization to external API using this grant flow when implementing fitness web services.

Client credentials grant is an authorization method which utilizes client credentials such as client ids and client secret ids to grant an access token for the client. Commonly no resource owners' data can be accessed to the clients by using the grant flow, unless the protected resources are under controls by the clients, or the protected resources were previously arranged with the authorization server. The common usage of client credentials grant is to improve the rate limit of data requesting to some resource servers, as fixed request limits may be preset in a fixed period of time for regulating their data traffics.

Not all the authorization grants are provided by the resource servers. For example, Fitbit services only provides authorization code grant and implicit grant flows for external parties to obtain authorization consents. Along with the access tokens granted by one of these grant methods, the clients are able to request protected resource from the resource servers in behalf of the resource owners.

4.4.3 Accessing protected resource by providing access token

In this phase, depending on which external REST API the client requesting to, various protected resource belongs to the resource owner can be fetched.

After successful authorization, the protected resource is hence accessible by appending the access token to the HTTP request. Figure 9 presents an example of requesting Fitbit activity data of the resource owner by HTTP GET request.

```
GET /1/user/-/activities/date/2015-03-01.json HTTP/1.1
   Authorization: Bearer eyJh….ROR5-o2wbN8t8eab9lbeeg
   Host: api.fitbit.com
   X-Target-URI: https://api.fitbit.com
   Connection: Keep-Alive
```

Figure 9: Authorized request to Fitbit server

The resource server then validates the access token and request scope, and responds the corresponding protected resource to the client. The Figure 10 presents the responding data returned in JSON format, based on the request described in Figure 9.

```
{
    "activities": [],
    "summary":
        {
            "activeScore": -1, "activityCalories" : 2457, "calorieEstimationMu": 2248,
            "caloriesBMR": 1665, "caloriesOut": 3678,"caloriesOutUnestimated": 3678,
            "distances": [], "elevation": 112.78, "fairlyActiveMinutes": 42,
            "floors": 37, "heartRateZones": [], "lightlyActiveMinutes": 359,
            "marginalCalories": 1497,"restingHeartRate": 69,
            "sedentaryMinutes": 603,"steps": 21653,
            "useEstimation": true, "veryActiveMinutes": 88
        }
}
```


Figure 10: Protected JSON data responded from Fitbit server

As it is shown from the data responded in Figure 10, many information is considered extremely personal. With the help of OAuth 2.0, accessing to a REST API and fetch data from the resource server is easy and secure.

4.5 Extensive concepts of web services

Although the traditional boundary of web services only covers the interoperating between web servers and a variety of web clients in different platforms, according to the definition given by Booth et al., web service can take place in different software applications and a variety of platforms [Booth et al., 2004]. Thus, applications built in platforms other than web clients are also considered as a part of web service. For example, mobile applications that requires Internet access, applications built in wearable platforms (such as watchOS) which directly or indirectly communicates with remote servers, should all be concluded as members of web services.

5. Implementation of Fitness web services

Comparing to other web services, there are two key characteristics in fitness web services: Firstly, based on the nature of fitness activities, keeping user engagement for the web service is harder than other services, due to the tendency of lacking motivation [Fankhauser, 2013]; Secondly, as fitness data is personal information, a proper method to secure the access of protected information is of importance for implementing robust and trust-worthy Fitness web services [Fitbit, 2017a].

Abiding these two characteristics, section 5.1 firstly introduces existing fitness APIs available in the market, implementing and maintaining by big and trustworthy companies. Based on whether the web services to-be-built use external Fitness API, section 5.2 presents the classification of fitness web services. Section 5.3 and section 5.4 respectively proposes the two categories based on the classification. Section 5.5 gives examples of fitness services orientating in different categories, and lastly section 5.6 generates guidelines for implementing Fitness web services integrated with gamification.

5.1 Existing Fitness APIs

Burgeoning fitness device providers have built their APIs for external access, especially with the increasing popularity of wearable devices in recent years. For example, Fitbit[15], Google Fit[16], Garmin[17], Nike Plus[18], and Runkeeper[19] have their REST APIs and documentations published. Thus, fitness data can be fetched from these service providers and response information can be displayed in a gamification way.

In order to access external data from remote servers, a secure and robust external API should be utilized for this process. In this thesis, the examples of famous fitness providers described above are assumed to have developed trustworthy REST APIs for external employment respectively.

For security consideration, registering the application and declaring usage scope in the fitness providers prior to using the REST API is obligatory. In addition, it is compulsory to follow the rules described in the provider's terms of service throughout the entire life span of the service. Due to the fact that all fitness related data from providers contains private information, one should not be able to access a user's data without the user's consent. Thus, it is highly recommendable to append authorization to the REST API of a Fitness service to protect the sensitive resources.

The format of the request URLs for the REST APIs can be found from the documentations composed by the service providers, usually recorded in their developer documentation websites. Besides, all accessible types of data available via the Web APIs are often listed in the documentation as well [Fitbit, 2017a].

---

[15] Fitbit API: https://dev.fitbit.com/
[16] Google fit API: https://developers.google.com/fit/
[17] Garmin API: https://developer.garmin.com/
[18] Nike+ API: https://developer.nike.com
[19] Runkeeper API: https://runkeeper.com/developer/healthgraph/overview

Most famous fitness service providers have provided the access of abundant protected and public resource which can be requested from their REST APIs. Amid the support of these providers, multiple points of end-user data can be stored, analyzed and displayed by the client. Table 6 presents the information of APIs of five fitness service providers, the information demonstrates that various of fitness data recorded by these fitness services can be requested from their REST APIs after authorization. However, as some of the service providers do not open client registration without some prerequisites, and some change their policies from time to time, it is recommended to do research and read through the terms and policies before integrating any external APIs.

Service provider	Data accessible	Client registry
Fitbit[15]	• Activity • Body & Weight • Devices • Food Logging • Friends • Heart Rate • Sleep Subscriptions	Yes, and free of charge
Google Fit[16]	• Managing Data Sources • Working with Datasets • Working with Sessions	Yes, and free of charge
Garmin[17]	• Speed • Distance • Pace • Calories • Cadence • Power • GPS • Time • Activities	Yes, $5,000 One-time License Fee
Nike plus[18]	• Activity • GPS Data • Experience type	Only open for partnership company
Health Graph by Run keeper[19]	• Profile • Settings • Fitness Activities • Strength Training Activities • Background Activities • Sleep • Nutrition • Weight • General Body Measurements • Diabetes Measurements • Personal Records • Friends • Comment Threads • Root Resource • Change Log	No longer accepting new Health Graph app registration as of the time composing of thesis.

Table 6. REST APIs information of five famous resource servers

5.2 Classification of fitness web services

As data is the core component in most Fitness service [Fankhauser, 2013], the architecture for building the fitness web services can differ a lot between services by the demands of external data. Thus, based on the usage of external Fitness APIs, two categories of fitness web services can be classified:

- Services accessing external data from external APIs,
- complete systems without utilizing external APIs.

Following sections (section 5.3 and section 5.4) will present integration of game-like elements, their respective advantages and limitations, and subcategories of these two categories.

5.3 Services accessing data from external APIs

Accessing fitness data from external APIs creates conveniences for implementation process, as it solves one of key characteristics of fitness web service: the access of secured data.

Depending on the requirements of a fitness web service, the architecture of the web service varies differently. Based on the service complexity and availability of external resource servers, three different types of services are classified. Subsection 5.3.1, subsection 5.3.2 and subsection 5.3.3 respectively present these three different categories, and subsection 5.3.4 discusses the relationship between these categories. Subsection 5.3.5 describes the advantages and limitations for building services accessing external data from external APIs.

5.3.1 Client-side only services

Client-side only services authorizing from resource server by implicit flow are simple web services which do not connect with databases and backend servers to storage protected data requested from resource servers.

The architecture of this kind of web services is simple: the service is running client side only by developing a front-end the view and handling the data traffic from the external API, by using front-end JavaScript controllers. Although this kind of services has several obvious limitations, the light-weighted and easy-implementation feature makes it beneficial for building small services and one-off services which only necessitated for specific events.

5.3.2 Solitary complete services

Solitary complete services are complete service fetches data from an external single resource

server using any kinds of authorization flow. A solitary complete service is a comprehensive system owning to the ability of storing data to its own database and hence build up its own API to external requests.

However, the solitary external resource server makes the protected data accessed purer, and it also grants the service ability to extend the functionalities. This kind of services are usually implemented as alternatives to the original fitness service providers, for example a web based game for Fitbit user.

5.3.3 Compound complete services

Compound complete services are complete services that fetch data from multiple resource servers. Compared to a solitary complete service, the compound complete service offers more possibilities of accessible data by building connections to numerous external resource servers.

However, the more external resource servers connected, the more complex the backend structure of the compound service will become. Figure 11 presents the web service *Yes.fit* which allows more than ten external fitness service providers integrated into its own service.

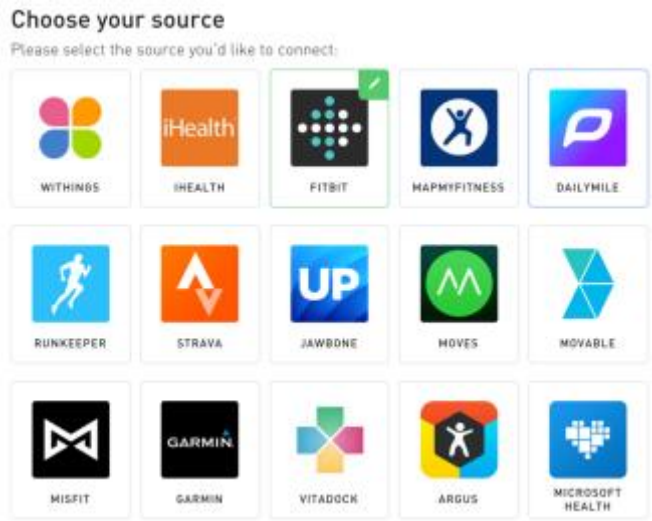

Figure 11. Yes.fit authorization page.

5.3.4 Relationship between different categories

The complexity of system structure is successively increased for these three types of services, and these services are upgradable from Client-side only services to Solitary complete services if backend systems are integrated, or from Solitary complete services to Compound complete services if additional external resource servers are added. Figure 12 presents the architecture and

interactions of these three types of services.

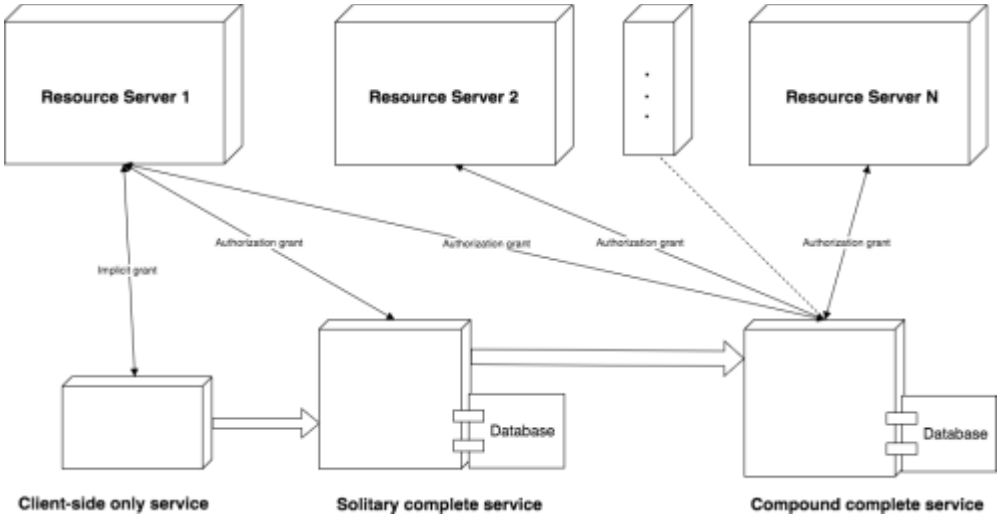

Figure 12. Architecture of three types of fitness web services.

5.3.5 Advantages and limitation

It is easy and secure to build a web service accessing external resources using OAuth protocol and REST APIs. Architecturally, the OAuth 2 protocol helps REST APIs address security concerns in a way that harmonizes the resource-centric manner and delivers stateless interactions with clients [Masse, 2011].

In addition, the structuralized nature of authorization flow of OAuth 2 makes implementation of the service fast, and the abundant resources provided from the REST APIs offer immense opportunities for gamification integration.

However, there are some potential problems for services which fetch data from external resource servers as they are highly depending on the API providers. For example, bad design or bad documentation of the structure of APIs by resource providers would strongly increase efforts for studying and implementing. Moreover, if the external resources make alteration of API structure or terms of use, it can tremendously impact the use of external APIs. Additionally, the potential broken connection of resource providers would prevent the data accessing and eventually destroy the user experience.

Some resource providers close the registration of client, or it charges expensively for the client registration, which make the external data totally not accessible.

5.4 Complete systems without using external APIs

In some circumstances, the use of external APIs is neglected or not allowed, for example due to the requirements and implementation scope. Thus, complete systems are built to fulfill these requirements and functionalities.

As it is introduced in Section 4.2, a complete architectural structure of modern web services includes front end servers, back end servers, and the communication between these two ends. Complete systems are services that built with the whole structure. Two main categories of complete systems are divided, subsection 5.2.1 and subsection 5.2.2 respectively present simple systems for specific events and generic systems with own fitness tracking methods, and subsection 5.2.3 summarizes the advantages and limitations of complete systems.

5.4.1 Simple systems for specific events

There are systems which are only designed and built for some specific events. The purposes of these kinds of fitness services are to motivate the fitness awareness of users, at the mean time to draw their attention to the designated events. A typical example is Olympics, which are international sporting events that consist of various competitions. The tradition of Olympics is to compete with top athletes from all around the world; hence, Olympic games are held every four years, and in each game, it is held in different host of cities and countries. In the forthcoming 2020 Tokyo Olympics, the host committee in Tokyo desires to not only draw people's attention for this Olympic game by the help of social media, but also to enhance user engagement and people's fitness awareness by building a gamification system [Miah, 2017].

When structuring these types of systems, of course it is possible to use external APIs if there is affiliation with external fitness service providers. However, if independence and self-governess are of top priority, building complete systems is a more approachable solution.

Simple is the key requirement for these systems. Therefore, the backend design does not need to be complicated, but it should include simple but robust database structure which can store user's credentials, basic information, fitness data, and the logical connectional algorithms for accurately processing communications.

The simplicity of backend design can be amended by providing good user interface design in the front end. This can be achieved by integration game-like elements to the system. For example, implementing *theme* element to a system makes the user interface attractive and novelty. Section 5.4.3 will introduce more details about the integration of game-like elements.

5.4.2 Generic systems with own fitness tracking methods

In section 5.1, the fitness service providers such as Fitbit[15], Garmin[17], Nike Plus[18], and Runkeeper[19] which have built their APIs for external access are mentioned. Due to the fact that they have their own fitness tracking methods (either wearable devices or mobile applications), the needs of business independence have limited them from accessing other fitness external APIs.

Comparing to simple system for specific events, and services accessing data from external APIs, the architecture of this type of generic systems is more complex and it requires more efforts to implement this kind of services, especially for building the back end of these services.

Similar to simple systems described in subsection 5.2.1, the backend servers for generic systems should handle strong encrypted data storage of user's credentials, basic information, fitness data is obligatory, as well as strong algorithm for interaction with front end clients. Besides, the ability of storing information of afflicted systems in the database is also necessary.

In addition, as it is likely for these services to distribute their own APIs for external access for conveying more business opportunities, designing robust APIs and securing the connection of API data flow by good authorization flow are of necessary. A robust design of REST API is recommended, as it not only enables the third-party applications to access the data, the front end of the own service can also fetch the system's own data with full access.

The frontend design of generic systems with own fitness tracking method can be similar to front ends of services accessing data from external APIs. However, there is more flexibility for generic systems, as data fetched are fully accessible and independent from any other services. Thus, more detailed data can be displayed in the front end of generic systems, and more choices for integrating game-like elements are available for this kind of systems.

5.4.3 Advantages and limitation

The advantage of complete systems is obvious: not relying on other external services, which insures the stability and independence of the designated systems, and enables the possibility for implementing any features and integrating any game-like element. However, this vastly weights the implementation efforts for building the system.

The independence characteristic of other external services sometimes can become a limitation for these complete systems, because it is hard to draw some users' attention if these users have already been tied by some other fitness services (for example, one may stick to the Fitbit ecosystem and skip all other similar fitness services).

However, the architectural structure of modern web services makes it easy to interchange

between different categories of systems. For example, if affiliation for external fitness service is added to a generic complete system, and the external data are accessed from the API provided by this external service, which will turn the system from a *generic system* to a *compound complete service.*

5.5 Examples of fitness services

In this section, three different web fitness services are listed as examples. Each of these examples shares the same core concept of utilizing gamification to promote fitness activities, but has its own unique purpose and background.

5.5.1 Trexplore

Trexplore is a web service which was targeting on an annual athletic event called *Finnkampen* in the summer of 2016. *Finnkampen* is a traditional athletics competition between Finland and Sweden which has lasted over 90 years [Ruotsiottelu, 2017]. Due to the competitive nature of this sport event, an online fitness service was requested by the Tampere city office to advocate citizen's fitness awareness, in addition to enable citizens from these two countries to participate in the *Finnkampen* event representing their own country.

5.5.2 Fitbit

Based on the health data gathered from Fitbit wearable products, Fitbit is a web service with several products, including Fitbit website, Fitbit mobile applications, and Fitbit wearable operation system. By using these products, users can check if they have reached their health goals in a fun, and empowered way [Fitbit, 2017b]. Because Fitbit have a huge lineup of wearable products which can collect user fitness and health data automatically and synchronize it to the database by mobile phone, some functionalities in Fitbit website differ from Fitbit mobile applications. Specifically, the main objective of Fitbit website is to display their fitness data in a broader and more visualized way, with some additional functionalities such as change user setting, online shopping and redeem rewards [Fitbit, 2017b].

5.5.3 Yes.fit

Yes.fit is first interactive virtual race platform in the world [Axial, 2016]. The web service aims at turning the boring fitness activities into virtual races, adventures, and fitness challenges, which as they have claimed [Intercom, 2017] will engage and entertain users while helping them reach their fitness goals and keeping them active during this process.

Yes.fit supports data accessed from multiple fitness applications and wearable devices, each time a user logs a workout or syncs his or her device, the fitness progress will update on the website. A virtual award is sent from the system once the goal has been reached by the user [Intercom, 2017].

5.6 Guidelines

This section focuses on generating guidelines on implementation of fitness web services and integration of gamification to these services, in order to build successful and meaningful fitness services which maintain user engagement and motivate fitness activities. The target audiences for these guidelines are not only software developers, but also investors who plan to order fitness services to comprehend deeper about this field.

Three steps are included in the guidelines, subsection 5.6.1 presents the first step *Identifying high-level requirements,* subsection 5.6.2 describes the second step *selecting suitable software architecture,* and subsection 5.6.3 explains the last step: *integrating game-like elements*.

5.6.1 Identify high-level requirements

Before integrating gamification to any services, requirement analysis is the first and most important stage to determine what functionalities and gamification are required by the service. This subsection presents some high-level requirements checkers for web fitness services:

Service platform: It is essential to decide what platform or platforms for the service to launch, as different platforms require completely different resources to implement, and they target on different user groups. Websites have been the safest choice for most web services, as they are scalable and can be accessed in various devices. However, since the popularity of mobile devices has gained recently, mobile applications and wearable device applications sometimes can be preferable approaches, if mobility is taken in consideration, or some mobile-only functionalities (for example, utilizing GPS or pedometer in the service) are required.

Programming languages: As it is described in section 4.2, even though server-side and client-side programming languages have distinctive syntaxes and unique grammatical structures, all languages can lead to the same outcome. However, decisions of programming languages should be settled in this stage. For example, software developer's preference and company's business plan can all be determining for choosing the best technologies. In addition, some modern programming languages have the ability to deliver cross-platform products, using these programming languages can significantly downscale the cost of implementation, if the web service plans to launch in several platforms.

Information storage: Since fitness services usually involve large amount of user information, it is recommended to determine if database is needed for the services.

Product life cycle: Not all services aim for running eternally; thus, estimating the product life cycle in this stage can help with structure selection and gamification integration in the future.

Project resource: Depending on the how many resources a project has possessed, the whole development process of this project can be affected.

External Fitness data integration: As it is introduced in earlier section, accessing fitness data from external APIs produces many conveniences for implementation process, but also raises some limitations for the service. If services do not desire any affiliation with external service, then it should omit external Fitness data integration; otherwise, it would be beneficial to have external Fitness APIs integrated to draw bigger user base, and simply the implementation process.

Service accessibility: This defines how do users access to a Fitness service. For example, is the service completely free to every user, or does it require purchasing the application or external devices prior to using this service.

Social expense: In the service, how often does and should a user interact with other users?

Expected service usage frequency: Expectation on how often does users use the service. Most fitness services desire user to be active in daily basis, however it is hard to achieve.

*Features keep users stick in the servic*e: As fitness activities usually demotivates average users fast. Apart from integrating game-like elements, fun and unique features can help with boosting the service usage, and keep users sticking in the service.

These requirement checks can help with identifying requirements of the designated service, depending on purposes of the services. Henceforth, software architecture of the services can be selected based on these high-level requirements.

5.6.2 Select suitable software architecture

Architecture selection of fitness services is related to the high-level requirements decision, mainly replying on *Information storage, product life cycle, project resource, external fitness data integration*, and *service accessibility*.

Services with limited *project resource* should refrain from using complex architectural structures, such as *compound complete services*, or *generic system*s with products in several platforms, as they usually require a lot of efforts to implement. On the other hand, services with big *accessibility plan* (For example, Fitness wearable providing services) should normally use

compound complete services or *generic system* architecture, in order to keep their independence and compatibility in the services.

If there is no external fitness provider involved in the requirements, services should be implemented as *complete systems*. Furthermore, if a system involves or has the potential of being external data provider, a generic complete system should be the system architecture; On the other hand, if a system is only designed for one or limited number of events (has short life cycle), and the resources for implementing the system is limited, then simple complete system is recommended. Table 7 presents the recommended selection of service architecture, based on the interactions between *information storage*, *product life cycle*, and *external fitness data integration*.

Information storage	Life cycle	External data integration	Recommended architecture
No	Short	No	-
No	Long	No	-
No	Short	Yes	Client-side only services
No	Long	Yes	Client-side only services
Yes	Short	No	Simple systems
Yes	Long	No	Generic systems
Yes	Short	Yes	Solitary complete services, or compound complete services
Yes	Long	Yes	Solitary complete services, or compound complete services

Table 7. Service architecture selection

5.6.3 Integrate gamification

Integrating gamification to services can be proceed, once requirements have been identified and system architecture has been selected for the services. Requirements of a fitness service decide which game-like elements should be integrated. For example, *social expense*, *expected service using frequency*, and *special features* described in subsection 5.6.1 all have decisive effects on

what and how gamification should be integrated into the service. This subsection demonstrates general integration of following common game-like elements in fitness web services.

PBL elements (*points, badges,* and *leaderboards*) are generally good for integrating, as they are proven to be effective for promoting motivational behaviors [Lister et al., 2014]. If the systems are based on user's interaction, *avatars, parallel communication, feedbacks*, and *peer pressure* are recommended elements to integrate. *Rewards, challenges, progress, levels,* and *clear goals* are generally applicable especially for generic systems, which Lister et al. have stated can potentially increase user engagement for fitness activities [2014].

Leaderboard: For client-side only services and solitary complete services, protected resources are fetched from only one external Fitness service provider. Thus, the leaderboard can be easily integrated by fetching the activities data of friends and activities data of self in a certain period of time (for example a week, or a month). After this, the same selected category of data will be compared between all entities and the results will be displayed in numeral orders. Commonly steps, distances, and even calories are selected for comparison. The procedure for integrating leaderboard to compound complete services is similar. However, as compound complete services involve data from several Fitness service providers, reviewing data from different sources should be taken into consideration. For example, it is possible that the naming rules of data categories differ between providers, or the data fetched from different sources actually indicate the same workout and hence doubles falsely the actual progress of the affected user.

Points and levels can be integrated to all services if the right formulas calculating the points are implemented. For example, *active points* can be counted by the total amount of active days, and *active tier* or *active level* can be determined based on the *active points* collected. Research is required in order to utilize the correct fitness data and formulas in representing the points and levels.

Achievements and badges: Similar to points and levels, achievements can also be integrated to all services from the accessed data. For example, *Marathon achievement* can be earned if the respond data include a single run farther than forty-two kilometers. However, it is disobeying the performance rule of the service if the service runs the same achievement determining code every time it fetches external data. Hence, if achievements and badges should be integrated for fitness web services, implementing complete services for data storage is recommended.

Goals and progress: It is difficult to keep track of user's goal and progress in client-side only services, unless the resources accessed from the external fitness service provider include as such information. In complete services, it is advisable to record the goals and progress data separately

in the server-side for maintainable reasons. Many fitness goals, such as weight goal, distance goal, and performance goal can be included in the service. Owning to the authorization grant, the latest weight data, or latest exercise activity which contains distance or performance data can be accessed automatically from the external resources server. Henceforth, the progress and the goal reach level of users can be calculated and updated in the server, and then be displayed in a lively way in the web service interface.

Challenge: It is achievable to integrate challenge without the implementation of data storage in client-only services, if the user interface of the services displays the challenge and performs the qualification calculation every time it fetches data from external resource severs. However, the users are not able to keep track of their completion progress to the challenge. Due to this reason, upgrading from client-only services to complete services is advisable to integrate challenge.

Story or theme: It is motivating for fitness services to have own stories or themes. All services are suitable for integrating storyline or theme element. For simple systems for specific events, *theme* or *story* is the key component of gamification in the systems, as the simplicity of the backend design require a strong theme or story to provide the attractiveness for users. On the contrary, due to the sufficiency of data provided by generic systems with own fitness tracking methods, basic game-like elements such as *avatars*, *goals* and *peer pressure* are good gamification approach for motivating users and increasing their engagement of fitness services.

Different from services accessing data from external APIs, the construction of complete systems does not reply on other external services. The benefit is that if necessary, every game-like element can be integrated to the systems.

6. Discussion

This chapter evaluates the guidelines proposed in this thesis. Section 6.1 presents validation of the guidelines using three different fitness services, and section 6.2 discusses the results and findings from the validation.

6.1 Validation of guidelines

The thesis chooses to apply the guidelines to three different fitness services, in order to ensure the validity of the validation process.

As the Trexplore is a project done by the author of this thesis, the requirements are approved to be accessed and displayed in this thesis. However, as Fitbit website and Yes.fit are commercial fitness services, their original implementation requirements are private and unable to access, but some high-level requirements can be assumed by their service descriptions in their respective websites [Fitbit, 2017b; Intercom, 2017].

6.1.1 Trexplore

According to the documentation of Trexplore project, the high-level requirement decisions for Trexplore are: building a website as the *service platform* which has database integration; the project predicts short product life cycle and has limited project resource; the service does not integrate with external fitness data and is accessible for everybody; there is low social expense needed in using the system, where it expects users to use the service daily.

Some key features of Trexplore are:

- User can register, login by email and password and find their credentials in case of forgetting password.
- User can log their daily activity anonymously; a maximum of ten kilometers or fitness activities equivalent of ten kilometers is limited daily.
- After logged in, user can log their daily activity a maximum of ten kilometers or fitness activities equivalent of ten kilometers is limited daily.
- After logged in, user can see his or her fitness data from all time, and different levels of trophies are given by their total distance.
- The total distances of two countries are shown as a form of match score in the index page.

According to guidelines from subsection 5.6.2, the trexplore service is recommended to build in the architecture of simple systems.

In the implementation of trexplore website, game-like elements such as *points, badges or achievements, leaderboards, levels, story or theme, clear goals, rewards, progress, avatars* are integrated to the system. The dashboard page as shown in Figure 13 presents *avatars, achievements, progress, clear goals* elements.

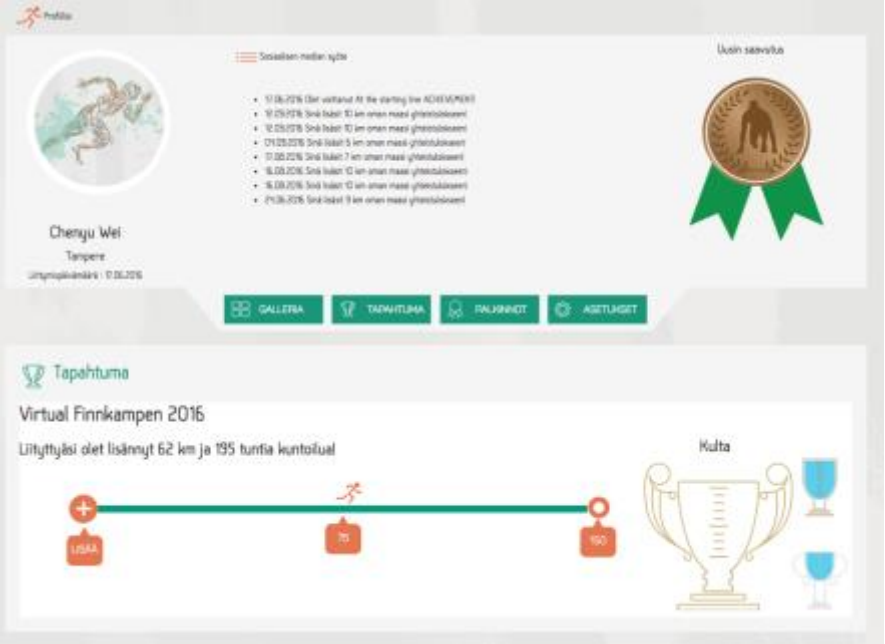

Figure 13. Trexplore dashboard page.

6.1.2 Fitbit

High-level requirement decisions for Fitbit are: building a website, iOS and android application, and wristband operation system as the *service platforms* which have solid database integration; The service plans to operate in a long run and has abundant project resource; In addition, this service does not integrate with external fitness data and is only accessible for Fitness device holders; Low social expense is needed for using the system, where daily activity is expected for user to use this service.

Key requirements of Fitbit website speculated by the thesis are:

- User can register, login by email and password and find their credentials in case of forgetting password.
- Display user's data gather from Fitbit wearable once the user is successfully logged in to the system.

- User can add, remove, and have social communication for other users.
- Challenges are provided by the system and can be created by users.
- Users can compare, complete their fitness data with others.

According to second step in the guidelines, Fitbit service is recommended to build in the architecture of generic systems.

Depending on the requirements, integrations of *leaderboard*, *achievement* or *badges*, *goals*, *progress*, *points*, *challenges*, *rewards*, *parallel communication*, *avatars*, and *peer pressure* are observed in the Fitbit website. Figure 14 presents the Fitbit website dashboard pages, where *points*, *progress*, *leaderboard*, and *avatars* can be identified.

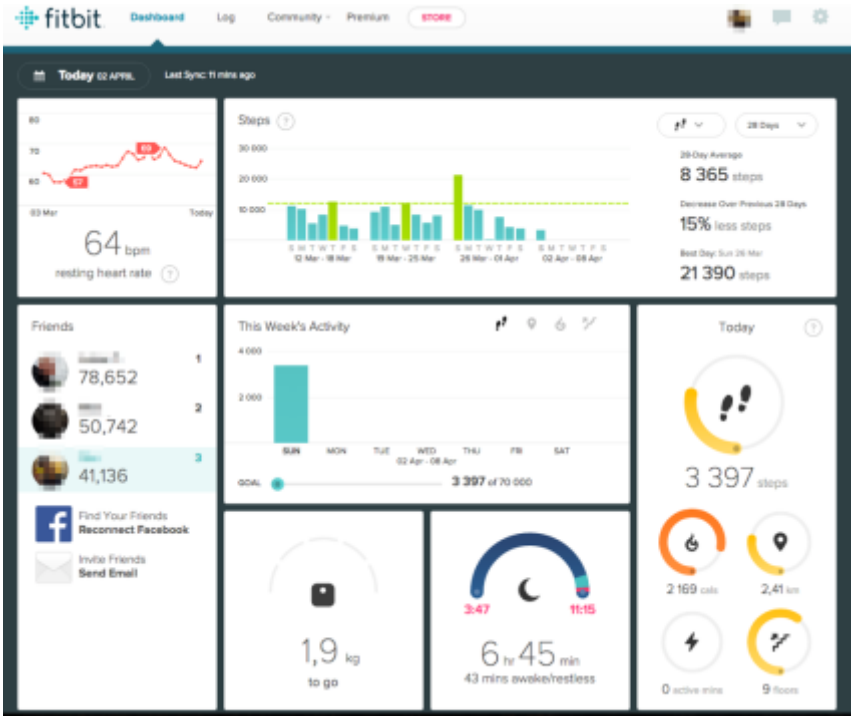

Figure 14. Fitbit dashboard page

6.1.3 Yes.fit

High-level requirement decisions for Yes.fit service are: building a website as the *service platform* which has good database integration and can access data from external fitness providers; The service plans to operate in a long run and has good project resource; The service is free to use for everybody; however, it may charge for extra contents; There is high social expense needed for using the system, and it expects users to use the service daily.

Key requirements of Yes.fit service speculated by the thesis are:

- User can register, login by email and password and find their credentials in case of forgetting password.
- User is able to authorize external fitness providers to the system, once succeed, their fitness data can be fetched automatically from authorized external service.
- Provide many virtual races reflecting real life routes, user will be given the responding award once the virtual race is finished.
- User is able to join a virtual race; the cost of different virtual races varies. Some of them are free, some of the races will charge for money.
- A secure system for handling payments.

According to second step in the guidelines, Yes.fit service is recommended to build in the architecture of compound complete service.

Observation of Yes.fit website indicates that *Points, badges or achievements, story or theme, rewards, challenges, progress, avatars,* and *clear goals* are integrated to Yes.fit service. Figure 15 presents Yes.fit dashboard, where only game-like elements *achievement* and *avatar* are displayed in this page.

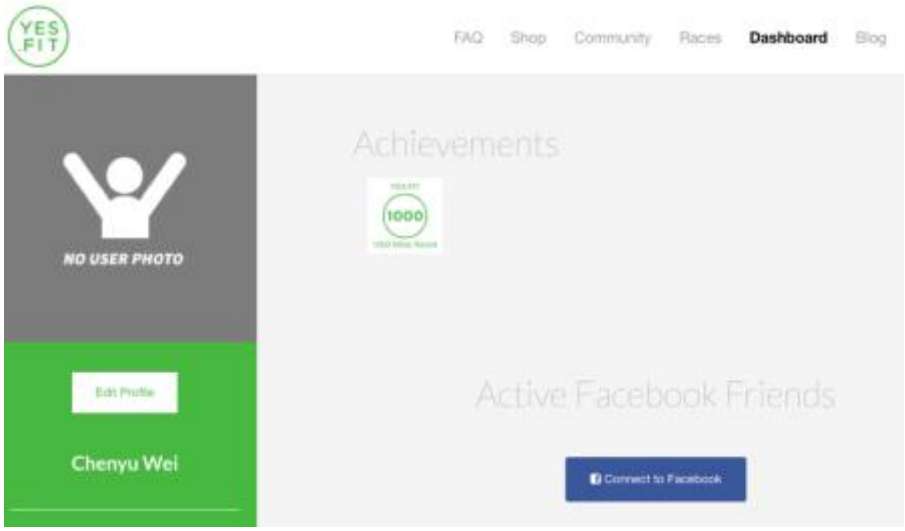

Figure 15. Yes.fit dashboard page.

6.2 Results

Identification of high-level requirements is quite easy for all three examples, even though the functional requirements and business goals for Fitbit and Yes.fit services are not accessible for this study. Table 8 presents different requirement decisions made by these three Fitness examples.

	Information storage	Product life cycle	Project resource	External fitness data integration	Service accessibility	Social expense	Expected service using frequency	Features keep users stick in the service
Trexp-lore	Yes	Short	Limit ed	No	Open for all	Low	Daily	See 6.1.1
Fitbit	Yes	Long	Abun dant	No	Only access to device owners	Low	Daily	See 6.1.2
Yes.fit	Yes	Long	Good	Yes, many	Open	High	Daily	See 6.1.3

Table 8. High-level requirement decisions

According to architectural observations of Trexplore website, Fitbit website and applications, and Yes.fit, it has proven that the architectures determined by the high-level requirement from the guidelines are equivalent to the real architectures used by these examples.

In addition, the gamification integration presents a promising effect on Fitness services. In the case examples, the coverage of integrating game-like elements to these three Fitness services are high. Table 9 presents Game-like elements utilized in three Fitness services.

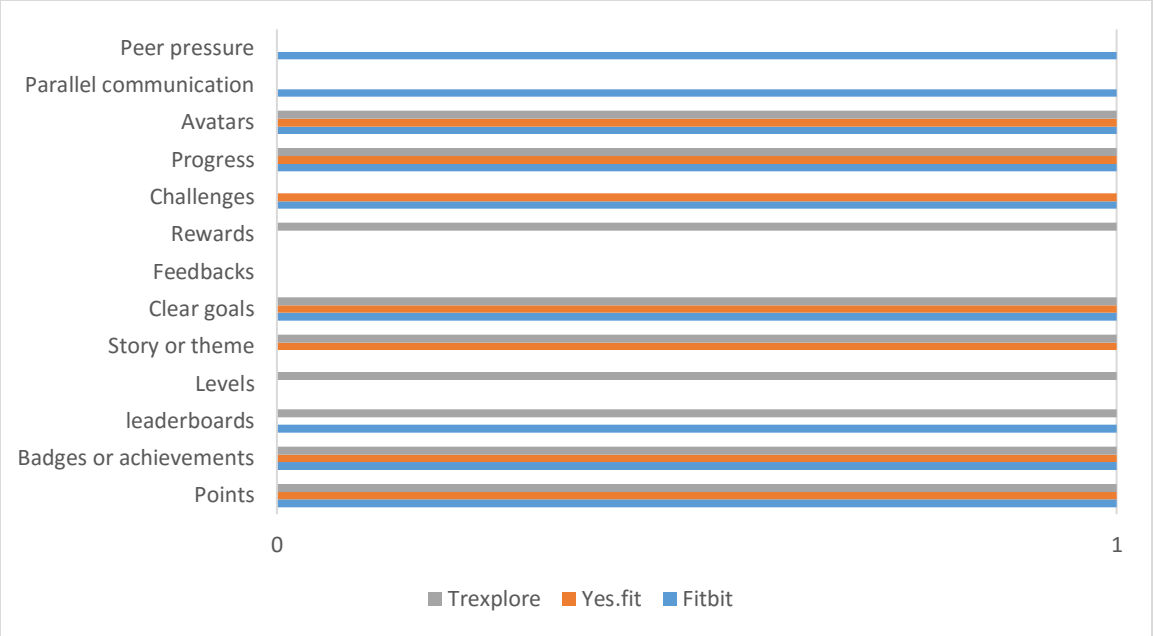

Table 9. Game-like elements identified in three Fitness services

In these three case examples, twelve out of thirteen game-like elements for gamifying Fitness services are identified. Among which, 61.5% elements are used in Fitbit service, 53.8% are used in Yes.fit service, and Trexplore utilizes 69% proposed elements.

7. Conclusion and future work

The goal of this research is to solve the vagueness in Fitness industry for implementing gamified services. There are two main research questions raised for this research: 1) How is gamification utilized in modern web fitness services? 2) What are the best practices for implementing fitness web services which integrate gamification elements?

Aiming at solving the research questions, this research analyzes the underlying components of gamification in the literature studies, and finalizes 13 game-like elements which are essential for Fitness services.

Subsequently, this research divides Fitness web services into two categories based on whether one or more external Fitness REST APIs are used for data accessing, namely services accessing data from external APIs and Complete systems without using external APIs. Furthermore, five subcategories (three for services accessing data from external APIs, and two for Complete systems without using external APIs) are introduced by the complexity of services and external resource servers.

Finally, with the results of finalized game-like elements and classification of Fitness web services, this thesis proposes guidelines for implementing fitness web services integrated with suitable gamification.

There is shortcoming in this research. For example, there are no empirical experiments used for validating the guidelines, which makes the research result a bit less convincing. However, the review results and discussion present a promising feedback toward these guidelines.

In the future, the research can carry on by using case studies to adequately validate the performance of the guidelines.

References

[Andra, 2017] Jacob Andra Apr 5, 2017. *A Good Addiction? How wearable devices and gamification can change healthcare*. Utah Business. http://www.utahbusiness.com/wearable-devices-gamification-change-healthcare/. Accessed in May, 2017.

[Attfield et al. 2011] Simon Attfield, Gabriella Kazai, Mounia Lalmas, and Benjamin Piwowarski. *Towards a science of user engagement* (position paper). In WSDM workshop on user modelling for Web applications, pp. 9-12. 2011.

[Axial, 2016] Yes.fit (previously named as MadeYesHappen) description in Axial. 2016. https://network.axial.net/a/company/ir4c-inc/. Accessed in May, 2017.

[Berners-Lee, 1989] Information Management: A Proposal. 1989. W3C.

[Bray et al., 2008] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, Sun Microsystems,. François Yergeau. *Extensible Markup Language (XML) 1.0* (Fifth Edition). W3C.

[Barry, 2017] Douglas K Barry. *Web Service: Service-Oriented Architecture (SOA) Definition*. http://www.service-architecture.com/articles/web-services/service-oriented_architecture_soa_definition.html. Accessed in June 2017.

[Bogost, 2011] Ian Bogost. 2011. *Persuasive Games: Exploitationware*. Gamasutra. http://www.gamasutra.com/view/feature/6366/persuasive_games_exploitationware.php. Accessed in May 2017.

[Booth et al., 2004] David Booth, Hugo Haas, Francis McCabe, Eric Newcomer, Michael Champion, Chris Ferris, and David Orchard. 2004. *Web Services Architecture*. W3C.

[Boller, 2013] Sharon Boller. 2013. *Learning Game Design: Game Elements to Consider*. http://www.theknowledgeguru.com/learning-game-design-elements/. Accessed in May 2017.

[Caillois, 1958] Roger Caillois. 1958. *Man, Play and Games.*

[Code, 2017] Codeschool. 2017. *Beginner's Guide to Web Development*. https://www.codeschool.com/beginners-guide-to-web-development. Accessed in June 2017.

[Csikszentmihalyi and Rathunde, 1993] Mihaly Csikszentmihalyi, Kevin Rathunde. 1993. The measurement of flow in everyday life: Toward a theory of emergent motivation. In J. E. Jacobs (Ed.), *Developmental perspectives on motivation* (pp. 57-97). Lincoln: University of Nebraska Press.

[Csikszentmihalyi, 1975] Mihaly Csikszentmihalyi. 1975. *Beyond Boredom and Anxiety*. Jossey-Bass Publishers.

[Deci and Ryan, 1985] Edward L. Deci, and Richard M. Ryan. 1985. *Intrinsic motivation and self-determination in human behavior*. New York: Plenum.

[Deci et al., 1999] Edward L. Deci, Richard Koestner, and Richard M. Ryan. 1999. *A Meta-Analytic Review of Experiments Examining the Effects of Extrinsic Rewards on Intrinsic Motivation*. Psychological Bulletin. 125, 6 (1999), 627-668.

[Deterding et al., 2011] Sebastian Deterding, Dan Dixon, Rilla Khaled, Lennart Nacke. 2011. *From Game Design Elements to Gamefulness: Defining "Gamification"*. MindTrek conference: Envisioning future media environments. ACM.

[Django, 2017] Django. 2017. *Django documentation*. https://docs.djangoproject.com/en/1.11/. Assessed in June 2017.

[Duggan and Shoup 2013] Kris Duggan and Kate Shoup. 2013. Types of rewards useful in business gamification. *Business Gamification For Dummies Cheat Sheet*.

[Engeli. 2004] Maia Engeli. 2004. *Review: Rules of play: game design fundamentals by Katie Salen and Eric Zimmerman. Leonardo*, Vol. 37, No. 5 (2004), pp. 414-415.

[Epstein, 1980] Joseph Epstein. 1980. *Ambition: The Secret Passion*. New York: E.P. Dutton, 1980.

[Fankhauser, 2013] Dani Fankhauser. 2013. *Is gamification just a fad?* Mashable. http://mashable.com/2013/05/17/gamification-buzzword/#nwrG1P0tOgqk. Accessed in May 2017.

[Fielding, 2000] Roy Thomas Fielding. 2000. *Architectural Styles and the Design of Network-based Software Architectures*. Doctoral dissertations, University of California, Irvine. http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

[Fitbit, 2017a] Fitbit developer guide. https://dev.fitbit.com/docs/oauth2/. Accessed in May 2017.

[Fitbit, 2017b] Fitbit about page. https://www.fitbit.com/fi/about. Accessed in May 2017.

[Flores 2015] Jorge Francisco Figueroa Flores. 2015. *Using Gamification to Enhance Second Language Learning*. Digital Education Review, (27), 32-54.

[Frodi et al., 1985] Ann Frodi, Lisa Bridges, and Wendy S. Grolnick. (1985). *Correlates of mastery-related behavior: A short-term longitudinal study of infants in their second year*. Child Development, 56, 1291-1298.

[Gassner, 2013] David Gassner. 2013. *Programming Foundations: Web Services*. Lynda.com from Linkedin.

[Graham, 1995] Ian S. Graham. 1995. *The HTML Sourcebook*. John Wiley & Sons, Inc., New York, NY, USA.

[Hamari and Eranti, 2011] Juho Hamari, and Veikko Eranti. 2011. *Framework for designing and evaluating game achievements*. In Proceedings of DiGRA 2011: Think Design Play, September 14-17. Hilversum, The Netherlands, pp. 122-134.

[Hamari et al., 2014] Juho Hamari, Jonna Koivisto, Harri Sarsa. *Does gamification work? — A literature review of empirical studies on gamification*. In proceedings of the 47th Hawaii International Conference on System Sciences, Hawaii, USA, January 6-9, 2014.

[Hamari, 2015] Juho Hamari. 2015. *Gamification: Motivations & Effects*. Doctoral dissertations, School of Business, Aalto University.

[Hardt, 2012] Dick Hardt. 2012. *The OAuth 2.0 Authorization Framework*. (Request for Comments: 6749) Internet Engineering Task Force (IETF).

[Huizinga, 1942] Johan Huizinga. 1942. *Homo Ludens*.

[Huotari and Hamari, 2012] Kai Huotari and Juho Hamari. 2012. *Defining gamification: a service marketing perspective*. In Proceeding of the 16th International Academic MindTrek Conference (MindTrek '12). ACM, New York, NY, USA, 17-22.

[Intercom, 2017] Yes.fit help website page. *Getting started with Yes.fit*. https://intercom.help/yesfit/general-info/getting-started-with-yesfit. Accessed in May, 2017.

[Jakobsson and Sotamaa, 2011] Mikael Jakobsson and Olli Sotamaa. 2011. *Special issue-game reward systems*. Game Studies 11, 1-2011.

[Järvinen, 2004] Aki Järvinen. 2004. *A meaningful read: rules of play*. Massachusetts: MIT Press. 670 pages.

[Kahn, 1990] William A. Kahn. 1990. *Psychological Conditions of Personal Engagement and Disengagement at Work*. The Academy of Management Journal. Vol. 33, No. 4, pp. 692-724.

[Kalin, 2013] Martin Kalin. 2013. *Java Web Services: Up and Running*, 2nd Edition. SafariBook.

[King et al., 2013] Dominic King, Felix Greaves, Christopher Exeter, Ara Darzi. 2013. *'Gamification': Influencing health behaviours with games*. Journal of the Royal Society of Medcine 2013. Vol 106, iss. 3.

[Kultima et al. 2015] Annakaisa Kultima, Kati Alha, Antti Syvänen, Heikki Tyni, Frans Mäyrä, Kati Koivu Timo Nummenmaa, Sami Serola and Klaus Törnkvist. 2014. *Rainbows, Unicorns and Hoverboards Making of OASIS Story of a dream at the University of Tampere*. ISBN 978-951-44-9607-3.

[Lazzaro, 2012] Nicole Lazzaro. 2012. *Gamification Can Kill*. Posted on 23rd March, 2012. http://www.nicolelazzaro.com/gamification/. Accessed in May 2017.

[Lenihan, 2012] David Lenihan. 2012. *Health games: A key component for the evolution of wellness programs*. Games for Health Journal 1(3), 233–235.

[Lister et al., 2014] Cameron Lister, Joshua H West, Ben Cannon, Tyler Sax, David Brodegard. 2014. *Just a Fad? Gamification in health and fitness apps*. JMIR Serious Games 2014, Vol. 2, iss. 2, e9, p.12.

[Lopez, 2011] Steve Lopez. 2011. *Disneyland workers answer to 'electronic whip'*. Los Angeles Times on October 19, 2011. http://articles.latimes.com/2011/oct/19/local/la-me-1019-lopez-disney-20111018. Accessed in May 2017.

[Malone, 1982] Thomas W. Malone. 1982. *Heuristics for designing enjoyable user interfaces: Lessons from computer games*. In Proceedings of the 1982 conference on Human factors in computing. systems (pp. 63–68). ACM.

[Marczewski, 2012] Andrzej Marczewski. 2012. *Gamification: A Simple Introduction*. (1st ed.). p. 3. ISBN 978-1-4717-9866-5.

[Masse, 2011] Mark Masse. 2011. *REST API Design Rulebook*: Designing Consistent RESTful Web Service Interfaces. ISBN: 978-1-449-31050-9. O'Reilly Media, Inc.

[Miah, 2017] Andy Miah. *Sport 2.0: Transforming Sports for a Digital World*. MIT Press, 2017.

[MuleSoft, 2016] MuleSoft. 2016. *The Top Six Microservices Patterns How to Choose the Right Architecture for Your Organization*.

[Nummenmaa et al., 2015] Timo Nummenmaa, Annakaisa Kultima, Ville Kankainen, Sampo Savolainen, Antti Syvänen, Kati Alha, and Frans Mäyrä. 2015. *OASIS deck of cards: house of colleagues: a playful experiment on community building*. In Proceedings of the 19th International Academic Mindtrek Conference (AcademicMindTrek '15). ACM, New York, NY, USA, 2-9.

[O'Reilly, 2005] Tim O'Reilly. 2005. What Is Web 2.0 - Design Patterns and Business Models for the Next Generation of Software. O'Reilly Media, Inc.

[Oren and Gal-Oz, 2014] Zuckerman Oren and Ayelet Gal-Oz. 2014. *Deconstructing gamification: evaluating the effectiveness of continuous measurement, virtual rewards, and social comparison for promoting physical activity*. Personal and Ubiquitous Computing 18.7: 1705-1719.

[Pereira et al. 2014] Pedro Pereira, Emília Duarte, Francisco Rebelo, Paulo Noriega. 2014. A *Review of Gamification for Health-Related Contexts. Lecture Notes in Computer Science.* Vol 8515.

[Rachel et al., 2016] Gawley Rachel, Carley Morrow, Herman Chan, and Richard Lindsay. 2016. *Gamification of Health Data from Fitbit® Activity Trackers. International Conference on IoT Technologies for HealthCare*. Springer, Cham.

[Reddit, 2017] Reddit Frequently Asked Questions. https://www.reddit.com/wiki/faq. Accessed in May 2017.

[Richterich, 2014] Annika Richterich. 2014. *'Karma, Precious Karma!' Karmawhoring on Reddit and the Front Page's Econometrisation*. Journal of Peer Production, 4.1.

[Robertson, 2010] Margaret Robertson, blog post *"Can't play wont play",* http://www.hideandseek.net/2010/10/06/cant-play-wont-play/. Accessed in May 2017.

[Ruotsiottelu, 2017] Ruotsiottelu. The finnish Finnkampen official site. http://www.ruotsiottelu.fi/. Accessed in May, 2017.

[Ryan and Deci, 2000] Richard M. Ryan, and Edward L. Deci. 2000. *Self-Determination Theory and the Facilitation of Intrinsic Motivation, Social Development, and Well-Being*. American Psychologist.

[Ryan and Grolnick, 1986] Richard M. Ryan, and Wendy S. Grolnick. (1986). Origins and pawns in the classroom: *Self-report and projective assessments of individual differences in children's perception*s. Journal Personality and Social Psychology. 550-558.

[Salen and Zimmerman, 2003] Katie Salen and Eric Zimmerman. 2003. *Rules of Play: Game Design Fundamentals*.

[Sara et al, 2003] Sara Price, Y Rogers. M Scaife, D. Stanton, H Neale. 2003. *Using 'tangibles' to promote novel forms of playful learning*. Interact Comput (2003) 15 (2): 169-185.

[Scheid, 2015] Stefanie Scheid. 2015. *Cultural and Personal Factors Affecting Mobile Language Learning – An Investigative Approach*. Master's thesis, School of Information Sciences, University of Tampere.

[Suits, 1978] Bernard Suits. 1978. *The Grasshopper: Games, Life, and Utopia*.

[Sutton-Smith, 1997] Brian Sutton-Smith. 1997. *The Ambiguity of Play*.

[Tamminen, 2015] Juhani Tamminen. 2015. *Gamification and User Engagement in Self-Learning Softwar*e. Master's thesis, School of Information Sciences, University of Tampere.

[Web, 2017] Web Technology surveys. 2017. Usage of server-side programming languages for websites: https://w3techs.com/technologies/overview/programming_language/all. Accessed in June, 2017.

[Werbach and Hunter, 2012] Kevin Werbach and Dan Hunter. 2012. *For the Win: How Game Thinking Can Revolutionize Your Business*. Wharton Digital Press.

[Werbach, 2017] Kevin Werbach. *Coursera: gamification*. Wharton School, University of Pennsylvania. Accessed in May 2017.

[Worddive, 2017] Worddive descriptive page. http://www.worddive.com/en/. Accessed in May 2017.

[Wu et al., 2012] Wu W, Dasgupta S, Ramirez EE, Peterson C, Norman GJ. 2012. *Classification accuracies of physical activities using smartphone motion sensors*. J Med Internet Res 2012.

[Xin et al., 2016] Tong Xin, Diane Gromala, Chris D. Shaw, and Amber Choo. 2016. *A Field Study: Evaluating Gamification Approaches for Promoting Physical Activity with Motivational Models of Behavior Changes*. International Conference on Human-Computer Interaction. Springer International Publishing.

[Zimmermann et al., 2004] Olaf Zimmermann, Sven Milinski, Michael Craes, and Frank Oellermann. 2004. Second generation web services-oriented architecture in production in the finance industry. In Companion to the 19th annual *ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications* (OOPSLA '04). ACM, New York, NY, USA, 283-289. DOI: http://dx.doi.org/10.1145/1028664.1028772