



OULUN YLIOPISTO
UNIVERSITY of OULU

Implementation and Analysis of a Support Tool for the Persuasive System Design Model

University of Oulu
Department of Information Processing
Science
Master's Thesis
Ashish Thapa
11.04.2016

Abstract

The primary objective of this study was to develop a novel artifact to provide categorized and ethical examples of design principles of Persuasive System Design (PSD) model. PSD model is considered as a fundamental design and analysis tool for creating systems that aim at behavior and attitude change ethically. With the increased attention in persuasive system design field and rise in Behavior Change Support Systems (BCSS), we can expect an increase in adoption of PSD model in design and development of such systems.

Despite these developments, examples of design principles from PSD model are neither readily available nor categorized. In response to this obstacle, we created and evaluated a support tool where users could share, obtain, and rate examples from each design principle. We utilized design principles from PSD model to implement the rating feature. We used Design Science Research Methodology (DSRM) to create the artifact. We tested desirability using Product Reaction Cards to understand user's feeling about the support tool. We also evaluated perceived effectiveness of both rating feature and app as a whole to identify the perceived effectiveness. From the results, we deduced intention to use and predicted actual usage in the future.

The support tool may be useful for persuasive system designers, researchers or beginners on this field to see real life examples of design principles of PSD model. Hopefully, it could support them on designing more ethical persuasive system or contribute towards the persuasive system design area.

Keywords

Support Tool, PSD model, BCSS

Supervisor

Professor Harri Oinas-Kukkonen

Dr. Piiastiina Tikka

Foreword

I would like to express my gratitude towards Professor Harri Oinas-Kukkonen for providing the opportunity and guidance to pursue my master's thesis. I must thank Piiastiina Tikka for being such an incredible supervisor and guide throughout this study. I am thankful to the members of OASIS for contributing time and effort in the evaluation.

A big salute to all the open source developers who put their effort for the betterment of technology with such open spirit. Finally, I would like to thank University of Oulu for providing this opportunity to pursue this thesis as a part of my degree.

Contents

Abstract	2
Foreword	3
Contents	4
1. Introduction	6
2. Existing knowledge and Motivation.....	7
2.1 Background	7
2.2 Related Works.....	8
2.2.1 Persuasive System Design Model	8
2.2.2 Behavior Change Support Systems (BCSS).....	10
2.2.3 Motivation	11
2.3 Ethical Issues	11
3. Research Method	13
3.1 Overview of Research Methods in IS	13
3.2 Design Science in Focus	13
3.3 Methodology for conducting Design Science Research	14
3.4 Implementation of DSRM in Support tool	15
4. Implementation and analysis	17
4.1 Persuasion Context Analysis	17
4.1.1 The Intent.....	17
4.1.2 The Event.....	17
4.1.3 The Strategy.....	18
4.2 Software Features	18
4.2.1 Rating System as a Feature.....	19
4.2.2 System Features and Functionalities	19
4.3 Software Artifact and its development process	20
4.3.1 Software requirement	20
4.3.2 Software architecture.....	21
4.3.3 Software Implementation	22
4.4 Artifact Demonstration	28
5. Evaluation.....	32
5.1 Research Setting	32
5.1.1 Objective of evaluation.....	32
5.1.2 Evaluation Participants.....	33
5.1.3 Evaluation method.....	34
5.2 Results.....	37
5.2.1 Results from desirability test.....	37
5.2.2 Results from Persuasive effectiveness evaluation.....	38
5.3 Findings	40
6. Discussion	43
7. Conclusion.....	45
References.....	47
Appendix A. Questions from Evaluation.....	52

SYMBOLS AND ABBREVIATIONS

Table 1 shows the abbreviations with their respective full forms used throughout the document.

Table 1. Abbreviations with full forms.

Abbreviation	Full Form
A-Change	Attitude Change
AWS	Amazon Web Services
API	Application Program Interface
B-Change	Behavior Change
BCSS	Behavior Change Support System
C-Change	Act of Compliance
CSS3	Cascade Style Sheet version 3
DSRM	Design Science Research Methodology
DWI	Design With Intent
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IS	Information System
MEAN	Mongo Express Angular Node
MongoDB	Mongo Database
MVC	Model-view-controller
O/C Matrix	Outcome Matrix
PDA	Personal Digital Assistant
PSD	Persuasive System Design
REST	Representation State Transfer
URL	Uniform Resource Locator

1. Introduction

The art of persuasion is not new. Kennedy (1994) mentions the inclusion of persuasion in early Greek history “Rhetoric”. B.J. Fogg, the initiator of persuasive technology, has stated that “persuasive technologies are ubiquitous; we are surrounded by digital products designed to change what we think and do” (Fogg, 2009). Oinas-Kukkonen and Harjumaa (2009) presented a systematic framework: Persuasive System Design (PSD) model to design and evaluate persuasive systems.

The PSD model describes the phases in persuasive system development, key issues, context, steps, and design principles. There are few applications using PSD model to create persuasive technologies such as Onnikka (Alahäivälä, Oinas-Kukkonen & Jokelainen, 2013), PerSoDeMetrics (Pribig & Felfernig, 2012), RiteOnTime (Tikka, Woldemicael, & Oinas-Kukkonen, 2016). Given the number of persuasive systems is on the rise, we identified the need for a system as a common platform. The system would provide readily available examples of PSD model, which could assist researchers, developers and other interested parties to collect, find, and review examples of persuasive systems features that follow the design principles of the PSD model. This study is an attempt to create a support tool for PSD model and evaluate it.

This is an interesting research because we try to utilize the principles of PSD model to design and analyze an artifact, which could potentially be instrumental in exploring the PSD model. The Design Science Research Methodology (DSRM) is utilized in this study. The reason behind this selection is that design science research embraces the idea of building artifact to understand and solve the problem-domain (Hevner, March, Park, & Ram, 2004), which closely resembles the purpose and method of this study.

The related studies in section 2.2 reviews and sheds light to the already existing knowledge. The artifact itself is the most important contribution of this study. The tool is a responsive web application containing a pool of example of PSD model and available (request for domain through <http://www.oasis oulu.fi/>) for public use. Anyone interested to view or contribute to the examples of PSD model can obtain or share information related to the persuasive system design model.

This article is divided into seven sections and each section is structured into one single chapter. The introduction section provides brief overview and lays foundation for the motivation of research. The second chapter portrays what are the achievements so far in persuasive system area and why this study is necessary. Third chapter discusses the available research methodologies and the reason for opting DSRM during this study. Fourth chapter describes the design and implementation part of the artifact. In the fifth chapter, artifact is evaluated. The discussion chapter depicts the results from the fifth chapter. In the last chapter, the accomplishments, drawbacks, and future roadmaps are discussed.

2. Existing knowledge and Motivation

This chapter unfolds some relevant contributions about art of persuasion to change behavior, its use in more recent times and ultimately in human computer interaction. The background section provides overview on origin and background of concepts of persuasive systems.

2.1 Background

Persuasion, according to Richard M. Perloff (2010) is, “a symbolic process in which communicators try to convince other people to change their attitudes or behaviors regarding an issue through the transmission of a message in an atmosphere of free choice”. Persuasion needs time and does not happen overnight. It should not include coercion as well. Instead, it is liberal way of providing the information and a democratic suggestion to make a right decision (Perloff, 2010).

Computers could as well be a technology to change attitudes and behavior through persuasion. Intentions behind the persuasion are always compelling. The functional triad, depicted in Figure 1, shows how computer technology could use persuasion to behavior/attitude change. It could act as a tool to perform activities, as a medium to convey the messages or as a social actor to create relationships and invoke social responses. Computers as persuasive technology, “Captology”, has emerged as a field of interest for researchers to understand and create ethical persuasive systems realizing the power and pitfalls (Fogg, 1999).

Fogg (2003) further mentions that different strategies should be implemented to persuade users depending on the techniques used. Which means each corner of the functional triad has own set of principles to be applied while using them. Fogg (2003) has identified seven persuasive technology tools: Reduction, Tunneling, Tailoring, Suggestion, Self-monitoring, Surveillance, and Conditioning. The reduction principle suggests use of technology to simplify complex tasks into simpler steps such that the user is persuaded to use the system. Tunneling provides user with the experience of an activity or a process. It should be conducted through a set of guided activities. However the set of activities should be ethical. Tailoring principle suggests that, persuasion in the system could be achieved by customizing the system as per the user requirements. Principle of suggestion requires the understanding of opportune moment to suggest so that it is more persuasive rather than disturbing or nagging the user in irrelevant circumstances. Self-monitoring principle empowers users to monitor their performance so that they could change/reinforce their attitude or behavior. Another principle is the surveillance or overt surveillance in this case, to observe the user’s behavior in order to shape the behavior or attitude. Conditioning principle suggests the use of technology to reinforce and shape positive behavior or transformation of the positive behavior into habits (Fogg, 2003).

Another corner of functional triad is computers as persuasive media. The principle of experience suggests that, by providing users an opportunity to get acquainted with a simulation environment, it is possible to change or shape user’s attitudes and behavior. He has outlined three ways to do so: cause and effect, simulated environment, and simulated objects. Cause and effect principle enables to observe the cause and effect of

simulation in real time and possibly persuade the user to change their attitude or behavior. The simulated environment allows user to rehearse their target behavior, motivate them through rewards, and persuade the behavior or attitude change. The simulated object lets the user understand the impact of behavior and attitude change in real world setting (Fogg, 2003).

The third corner of the functional triad is computers as social actors. The computer creates social relationships with users. Then it uses those relationships to persuade the users to attain attitude/behavior change. The system could use physical aspects such as physical appearance or body movement to persuade the user. Similarly, it could use psychological aspects such as feelings, humor or language (sign or spoken) for persuasion. It could also use social rules or roles for persuasion. In a nutshell, any system designed with the intent to change user's behavior or attitudes could be identified as persuasive systems (Fogg, 2003).

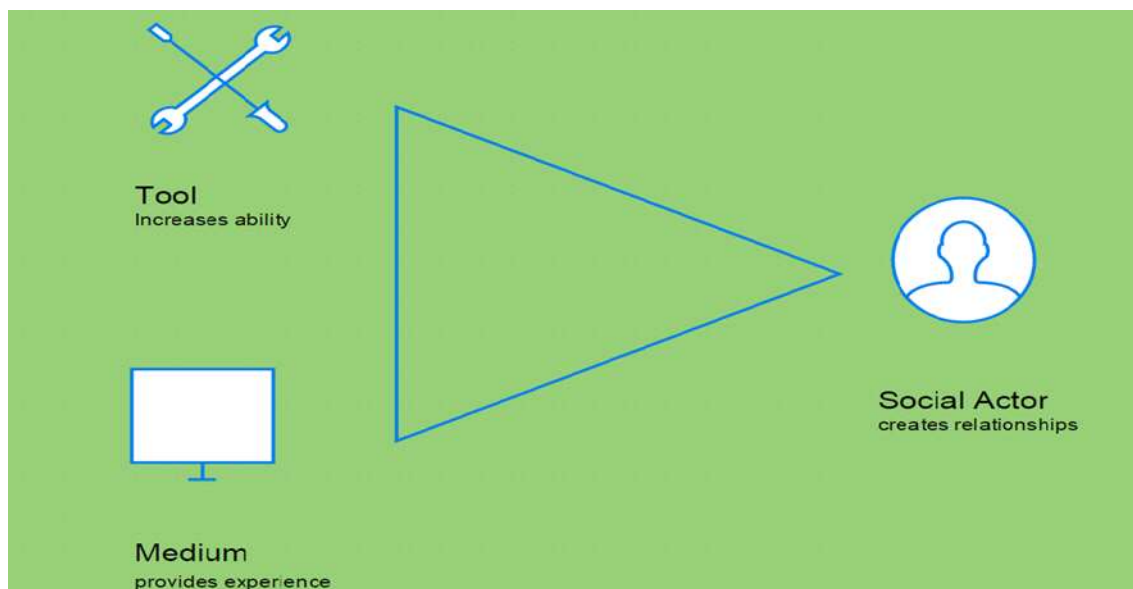


Figure 1. Functional Triad (cf. Fogg, 1999, p. 28)

2.2 Related Works

This section explains the prior literature and recent efforts, which led to the conception of this study. The concept of support tool for persuasive systems actualizes from the origin of the PSD model. Section 2.2.1 illustrates PSD model in broader sense. Section 2.2.2 explains BCSS, which are currently the primary target systems for implementing PSD model including the proposed architecture by Alahäivälä et al. (2013). Section 2.2.3 analyzes the motivation behind the support tool and finally, section 2.2.4 puts forward some ethical issues mentioned by persuasive systems experts.

2.2.1 Persuasive System Design Model

The persuasive system design involves three steps: understanding key issues behind persuasive systems, analyzing the persuasion context, and finally designing the system qualities according to the persuasive system design principles (Oinas-Kukkonen & Harjumaa, 2009). Figure 2 is a visual representation of PSD model and illustrates the phases that lead to behavior/attitude change.

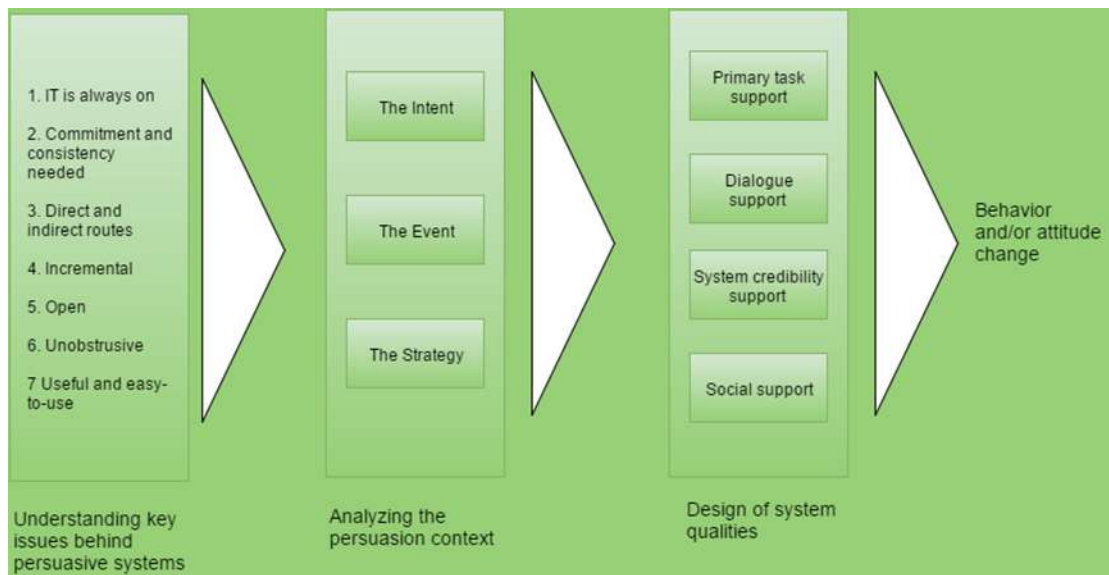


Figure 2. Phases in Persuasive System Development (cf. Oinas-Kukkonen & Harjumaa, 2009, p. 487)

In the first step, the seven postulates from the PSD model should be addressed while designing/evaluating persuasive system. The postulates could be summarized as: IT is actively persuading. Commitment could be used as technique to persuade the user. Persuasive strategies should be customized according to the recipient's nature. The process of persuasion works effectively if the process is gradual. The system should be unobtrusive, user-friendly, and effective (Oinas-Kukkonen & Harjumaa, 2009).

The second step, analyzing the persuasion context, requires precise, comprehensive understanding and distinction among persuasion intent, event, and strategies. It is necessary to identify the intent, the persuader, and the change type to define the intent of persuasion. The event could be analyzed using *use context* (problem domain related features), *user context* (user related feature such as goals, motivation, lifestyles etc.), and *technology context* (technology dependent features such as availability, strength and weakness, reliability). The strategy requires understanding about the user and deciding the persuasion route to achieve behavior/attitude change. The final stage includes designing the system qualities. It is a three-step process where the first step is to analyze the persuasion context and select the suitable design principle. The second step is to define the requirements. And the third step is to implement the software (Oinas-Kukkonen & Harjumaa, 2009).

The design principles are categorized into four groups and act as guidelines to discover requirements for software qualities. Each category serves a unique purpose. For example, the design principles which help to perform user's primary task are categorized under Primary Task Category. The primary tasks could be carried out by reduction, tunneling, tailoring, personalization, self-monitoring, simulation, or rehearsal. Which means the persuasiveness could be introduced into the system or evaluated using above mentioned ways. Similarly, Persuasiveness in the system could be introduced by dialogue between system and the user. The dialogue support category includes computer-human dialogue related principles. This category comprises of praise, rewards, reminders, suggestion, similarity, liking, and social role. Having credibility in the system also increases persuasiveness in the system. Trustworthiness, expertise, surface-credibility, real world feel, authority, third party endorsements, and verifiability are included in system credibility as they exhibit credibility traits. Social support category comprises of design principles, which use social attributes to increase persuasiveness in the system. It includes social learning, social comparison, normative

influence, social facilitation, cooperation, competition, and recognition (Oinas-Kukkonen & Harjumaa, 2009).

2.2.2 Behavior Change Support Systems (BCSS)

According to Oinas-Kukkonen (2010), “A behavior change support system (BCSS) is an information system designed to form, alter or reinforce attitudes, behavior or an act of complying without use of deception, coercion or inducements” (Oinas-Kukkonen, 2010a, p. 6). The changes could be identified as act of compliance, behavior change, and attitude change represented by C-Change, B-Change, and A-Change respectively. The order of difficulty is also respective. It means, C-Change is usually the easiest because the user responds to a single request for compliance and does not expect any behavioral change. B-Change is relatively difficult than C-Change as it tries to change the user’s behavior. A-Change is the most difficult to attain because it requires extremely high motivation from the user and sustainable behavior change. The change can also be represented in terms of formation, alteration, and reinforcement of the behavioral change. Reinforcing the behavioral act is easier compared to both forming and altering behavioral change. Forming is moderately difficult whereas altering the behavioral change is the most difficult to attain (Oinas-Kukkonen, 2010a).

In the same article, Oinas-Kukkonen mentioned that, “The Persuasive System Design model is the state of the art conceptualization for designing and developing BCSS” (Oinas-Kukkonen, 2010a, p. 8). The layered architecture suggested by Alahäivälä et al. (2013) incorporates behavior change support system with PSD model. This architecture enables the designers to understand key system components. They have identified 5 main components of persuasion context. Figure 3 represents those components followed by brief description of each.

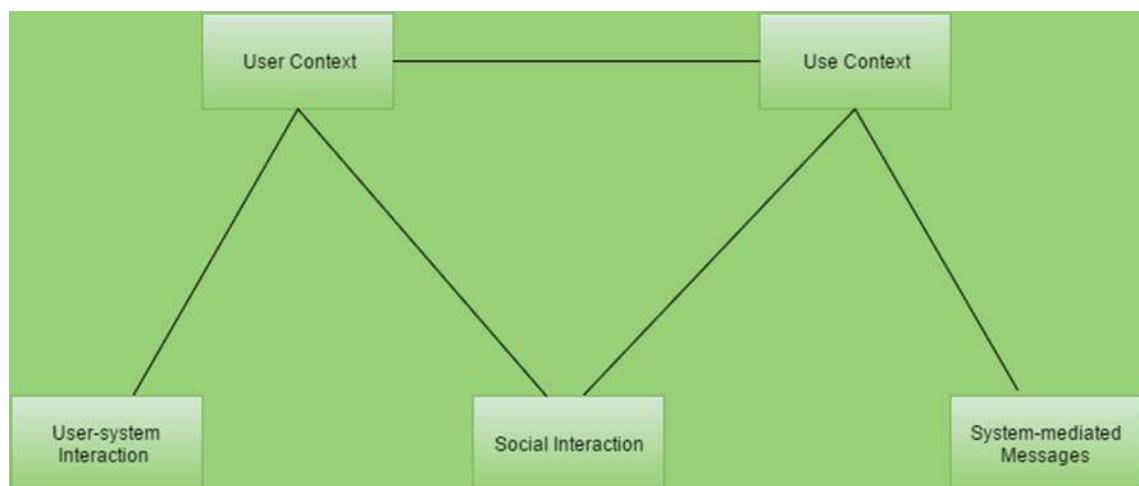


Figure 3. Architectural components of persuasion context (cf. Alahäivälä et al., 2013, p. 6)

The user context component should comprise those features containing information on user, his/her needs, motivation etc. The use context feature should be responsible for responding to the problem domain by applying fitting strategy for persuasion. User-system interaction, as the name suggests, enables the interaction between system and the user to set targets and find ways to meet them. Social interaction provides the user an opportunity to interact with desired people to receive and provide information or support. System mediated messages are the components responsible for behavior change if used in use context properly. Case Onnikka is an example actualizing the architectural components represented in figure B following the PSD model and process.

(Alahäivälä et al., 2013). Another effort to utilize PSD model is recognizing software design pattern for social influence by Oduor, Alahäivälä & Oinas-Kukkonen (2014). It attempts to provide general techniques to be used while designing and implementing persuasive systems by using principles from social support category.

2.2.3 Motivation

There has been substantial growth in the use of behavior change support systems. The O/C matrix and PSD model support in designing such systems (Oinas-Kukkonen, 2010b; Oinas-Kukkonen, 2013). An example from Fogg (2003) shows that there are not enough examples of persuasion. It is apparent that people might be misled or misinformed about the persuasive technology (Fogg, 2003). The current and future scenarios suggest that the designers should have thorough understanding on the persuasion to use it in their system (Räisänen, Lehto, & Oinas-Kukkonen, 2010).

The findings from Räisänen et al. (2010) suggest that, despite the interrelation between persuasive system design principles, there is no guarantee that every combination of principles would work efficiently. Instead, the techniques should be selected on the basis of coherence and compatibility to serve the primary task. They have acknowledged the overlapping and interlink between the designs principles in PSD model as a challenge. The major downfall is the lack of heuristic evaluation of the PSD model as it may increase the potential bias emerging from subjective views of the designer (Räisänen et al., 2010). An expert evaluation (Langrial, Lehto, Oinas-Kukkonen, Harjumaa, & Karppinen, 2012) of persuasive features shows the need for improvement in the implementation of persuasive features. It also mentions that some of the principles are ambiguous (Langrial et al., 2012).

It becomes evident that there is a need for more practical examples (Fogg, 2003; Räisänen et al., 2010) so that interested people could get idea on what are persuasive techniques and what are not. We already have the design principles from the PSD model but we lack a gallery of categorized instances. The PSD support tool is an effort to bring the design principles to life by collecting and curating examples of PSD design principles.

Each persuasive category would have the design principles and their respective examples. The support tool has usage in different areas. For example, the designers could familiarize themselves with different persuasive design principles and their examples. The students and researchers could use the data for their study purpose. And the persuasive study area could have a single place to gather practical information involving active participation from related field. This may increase the radius of the subject area and interest from the public.

Furthermore, the support tool allows readers to rate the article uploaded by other users. Articles with the best reviews are featured at the top of list for each design principles. The reason behind this feature in the support tool is to persuade the users to find genuine and relevant examples of the principles.

2.3 Ethical Issues

Persuasive technology may not always motivate users to act rationally or ethically. There are possibilities of unethical persuasion: incidental, accidental or even planned. Persuaders, usually the designers of system, should be aware and responsible for the outcome. Most researchers agree to the idea that, ethical issues need to be addressed while creating persuasive systems (Berdichevsky & Neuenschwander, 1999; Fogg,

2003; Oinas-Kukkonen, 2010a; Oinas-Kukkonen, 2010b; Smids, 2012; Verbeek, 2006). We do not have a silver bullet to eliminate all ethical issues while designing persuasive systems (Berdichevsky & Neuenschwander, 1999; Fogg, 2003; Karppinen & Oinas-kukkonen, 2013; Verbeek, 2006.) but there have been some interesting advancements. The framework of ethical approaches in design of BCSS suggests three approaches: a) Guideline, b) Stakeholder analysis and c) User involvement approaches where voluntary change in behavior and transparency in the persuasion are the key issues (Karppinen & Oinas-Kukkonen, 2013). The design with intent (DWI) is a framework which enables the designers to understand the user behavior in the early design phases. It tries to solve the problem of random or unintended change that the system might induce (Lockton, Harrison, & Stanton, 2010). However, we still do not have a universally accepted standard to create outright ethical BCSS or persuasive systems.

3. Research Method

First section in this chapter provides a general overview of the available research methods in Information systems. Second section analyzes Design Science Research Method. Third section explains the DSRM. The final section illustrates the process and adoption of DSRM in our research context.

3.1 Overview of Research Methods in IS

Majority of research in Information Systems conducted so far can either be classified into behavioral science paradigm or design science paradigm (Hevner et al., 2004). According to Hevner et al. (2004), behavioral science emerged from natural science. It is theory building and justifying the theory by nature. This research paradigm focuses in understanding and analyzing the individual or organizational phenomena during the research process (Hevner et al., 2004; March & Smith 1995).

On the contrary, design science research is based on engineering (Hevner et al., 2004; Simon, 1988). “The design-science research paradigm seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts” (Hevner et al., 2004, p. 75). To perform relevant and effective research, it is essential that both these natural and design science concepts should be addressed (Hevner et al., 2004; March & Smith 1995).

3.2 Design Science in Focus

Hevner et al. (2004) provided a framework to illustrate a better way to conduct, evaluate and present the design science research. The Information system research framework (Hevner et al., 2004) outlines the boundaries and provides guidelines for better design science research. This framework consists of three major components: Environment, IS Research, and Knowledge Base. The knowledge base comprises of Foundation and methodologies to realize the rigor. Foundation mostly includes existing theories, frameworks and design artifacts. The methodologies provide guidelines for evaluation or justification of artifacts or theories. The research is conducted to either build and evaluate the artifacts or develop and justify the theories. The business needs are identified from the environment that consists of people, organization and technology (Hevner et al., 2004).

The process of building an artifact and applying it to solve an existing problem enables the designers to understand the design problem extensively. Comprehensive understanding of problem reveals effective ways to solve them. Based on this principle, Hevner et al. (2004) have derived seven guidelines that could aid researchers to perform design science research more effectively. These guidelines aim to provide better understanding of the requirements to be effective in the research. Hevner et al. (2004) emphasize in using the seven guidelines in order to make the research complete however they urge not to use the guidelines just for the sake of completion. The first guideline and probably the most important one is *Design as an artifact*. It expects the outcome of a design science research to be an artifact. The resulted artifact should be novel and should be able to solve problem domain in innovative way. Designing and developing an artifact only does not fulfill this guideline. The presentation of artifact

should also be clear and easy to understand. The description should aid in adoption and implementation of artifact. Instantiations, construct, models and methods could be a representation of IT artifacts as they attempt to solve different problems of various domains (Hevner et al., 2004).

The second guideline is *Problem relevance*. It recommends that the designed artifacts should aim to solve the problems faced by the information systems community. The artifacts should be designed so that they could be effective in solving relevant problems rather than trying to prove existing theories or predicting new phenomena (Hevner et al., 2004). The third guideline is *Design Evaluation*. It asserts evaluation as a key component in design science research. The evaluation techniques adopted should be utilized properly and the artifact should undergo rigorous evaluation to test its utility, quality and efficacy. Design itself is a cycle which includes creating some artifact, evaluating it and using the feedbacks to solve the shortcomings from the design. It is important to realize that selection of evaluation techniques should be done according to evaluation metrics and artifacts designed (Hevner et al., 2004).

Hevner et al. (2004) have mentioned *Research Contribution* as fourth guideline where the artifacts must be novel and innovative. It should attempt to solve an unsolved problem. The research must add value in design artifact, foundation or methodologies. *Research Rigor* is the fifth guideline, which we could relate to the third guideline as well. Design science research should utilize rigorous methods formulated from existing knowledge base in both design and evaluation activities. It is crucial to realize the balance between rigor and relevance. Inclusion of either one should not result into exclusion of another.

The sixth guideline of Hevner et al. (2004) *Design as a Search Process* states that design is an effort to attain the best possible solution to an existing problem. Since design is an iterative process, construction, evaluation and improvements cycle should be used in iteration to obtain desired artifact. The final guideline, *Communication of Research*, emphasizes on communication and presentation of artifact in organized and effective way. It is necessary to understand the target audience and communicate accordingly. The research and artifacts may not be worth much if it is not presented and communicated properly. To summarize, “In the design-science paradigm, knowledge and understanding of a problem domain and its solution are achieved in the building and application of the designed artifact” (Hevner et al., 2004, p. 75).

Peppers et al. (2007) advocate that the design science research could have advanced quite a bit, had there been a methodology along with commonly accepted framework for research purpose. They have proposed a methodology commonly known as Design Science Research Methodology (DSRM) to solve this vacuum. The following section provides short analysis about the methodology and its usage in our study.

3.3 Methodology for conducting Design Science Research

The Design Science Research Methodology proffered by Peppers et al. (2007) aims to assist in performing research and presenting the outcome in an uncomplicated way. The use of framework allows successful research whereas the mental model enables better presentation. It consists of 3P's i.e. principles, practice and procedure to perform the research. The DSRM has to meet following 3 bare minimum objectives:

- Consistency with prior literature
- Provide Nominal model to perform Design Science Research
- Provide Mental model to present and evaluate Design Science Research

Peffers et al. (2007), identified 6 activities that constituted the model. The DSRM illustrates 6 activities along with possible entry points. The following paragraph explains each activities of the model in nutshell.

The first activity is *Problem identification and motivation*. It focuses on specific research area and the need for a particular research. If the research problem is not clearly identified, it will usually lead to false direction with most probably wrong results. It is the most important activity and should be meticulously performed. The second activity is *Define the objectives for a solution*. This activity allows researchers to draw boundary of research area and target the achievable. The contemporary developments and studies help to narrow down the scope. The third activity is *Design and development*. This activity involves identification of functional and nonfunctional requirements. These requirements are then transformed into design and use the design to build an artifact. The artifacts in this context could be a construct, model, method, or instantiation. The fourth activity is *Demonstration*. It is the phase after the artifact is constructed. But the artifact does not have to be polished and ready-to-ship. It could be shown in simulation, as a proof-of-concept or akin. The fifth activity is *Evaluation*. It demonstrates how well the artifact developed achieves the solution objective. The form of evaluation varies with the problem area. A single evaluation technique may not guarantee the correct representation of the entire problem domain. So it becomes a prerequisite to know the relevant metrics and techniques. The sixth activity is *Communication*. The study is meaningless if the result of study does not reach the targeted audience. All the hard work and novelty won't have much effect unless the message is not properly delivered. The researcher should be aware of the structure which might vary according to the discipline (Peffer et. al, 2007).

3.4 Implementation of DSRM in Support tool

Approach utilized: Problem centered approach was selected as the entry point since it attempts to solve an existing problem of unavailability of examples for PSD principles. The PSD model (Harjumaa & Oinas-Kukkonen, 2008) and the adoption of this model including BCSS (Oinas-Kukkonen, 2010a) provides the basis for this study.

Problem Identification and motivation: The number of health related persuasive systems have increased dramatically and the system to support behavior change are increasing (Alahäivälä et al., 2013; Intille, 2004; Kraft, Drozd, & Olsen, 2009; Purpura, Schwanda, Williams, Stubler, & Sengers, 2011). There is a need for study on effective design techniques to address this topic (Kraft et al., 2009). The persuasive system design model is a great resource to design and develop behavior change support systems (Lehto & Oinas-Kukkonen, 2011). However, one cannot find the examples regarding the PSD model. The need for system was realized where one could get all the PSD related literature and examples.

Objectives of the Solution: The objective was to develop a support tool where users could obtain or share PSD model related examples and provide their opinion through rating system. The system was conceived to be a social site for people who wish to share and learn more about PSD technologies. The examples would be contributed from the registered user and categorized according to the design principles specified in PSD model.

Design and Development: The functional and nonfunctional requirements were identified through formal software engineering principles. The requirements were rigorously analyzed and frequently discussed. The 3-tier architecture was used for abstraction of different layers. The modern web technologies were used and MVC

design pattern was used to separate logic from views. The system has been comprehensively documented for future reference and modification. Since the system is related to PSD model, the postulates and design principles were used wherever applicable.

Demonstration: Initial release was made for the evaluation purpose and the beta version is available for public use. The evaluation section explains the questionnaire, evaluation process and result. The artifact is demonstrated in chapter 4.3.4 of this thesis. It includes the screenshots and description of the implemented system. The screenshot shows certain parts of the application and description depicts the detail about that particular section.

Evaluation: The evaluation was an interesting mix of heuristic evaluation and persuasiveness testing. It involved seasoned experts of persuasive system design. In addition to extensive knowledge on PSD model, the experts had expertise on usability or software development or both.

Communication: Direct communication is done through this thesis. It includes the design artifact, methods utilized for research and evaluation and the discussion about the drawbacks or possible improvements along with the author's remark about the study. Indirect communication could be the use in future research papers if someone is interested to utilize this study in their academic or industrial work.

4. Implementation and analysis

This chapter discusses the implementation and analysis of the software artifact. It reviews the context in which the artifact was developed. Also, it sheds light on the technical aspects of the artifact. This support tool could be beneficial for anyone looking for concrete examples of persuasive system design. The artifact itself is not a BCSS however, it possesses some persuasive features. For that reason, we have attempted to use PSD model as a guiding framework to analyze the system.

The support tool is available upon request (request for domain through <http://www.oasis oulu.fi/>). This tool provides opportunity for registered users to contribute to the large pool of data and get their posts featured in the list.

4.1 Persuasion Context Analysis

After understanding the basic issues behind the persuasive systems, the next major step is to understand and analyze the persuasion context. This following section demonstrates analysis of the intent, event, and strategy (see section 2.2.1 for more information on PSD model and its stages).

4.1.1 The Intent

It is necessary to identify persuader and the type of change to fully understand the intent (Oinas-Kukkonen & Harjumaa, 2009). In our case, the designer of the support tool is the persuader. The intent of the system is to persuade users to contribute the content, i.e. examples of PSD principles and learn from it. The section 2.3 covers motivation in detail. The intent of the support tool is to provide a single solution where people can obtain and contribute the data that are relevant in designing persuasive systems. Although there is a growing interest in persuasive systems, there seem to be a lack of agreement on what persuasive features are. The lack of real-life examples is also acting as a roadblock for designers with limited PSD knowledge from using it. Using this support tool, we expect even the beginner of PSD system to be able to apply principles and design the system with ease. The user once becomes more informed about the principles has an option to share their knowledge back to the community.

4.1.2 The Event

User Context: The support tool is targeted from novices to the experts of the PSD model. It could be especially useful for designers looking for examples of PSD principles to implement in persuasive systems. The users would benefit by getting the necessary information about different sections of PSD model at one place with examples monitored by the experienced and motivated personnel. It is also suited for the people who refrain from using proprietary tools. The posts could be rated by the public and the highest rated posts are featured at the top. Users can openly access the information or rate other's posts. In future there is a possibility for the user to be acknowledged as a moderator.

Use Context: The support tool manifests use case for multiple user groups. The most important being real-world examples from other fellow contributors, getting response and sharing knowledge with the community.

Technology Context: The tool is available for use in most modern browsers and different screen sizes. The accessing device could range from small handheld devices such as mobile, PDAs, and tablet to laptops or desktops. To remain accessible and competitive with modern technologies, we implemented HTML5, CSS3, and JavaScript for views, styling and scripting respectively.

Bootstrap (Bootstrap, n.d.) was used along with HTML5 for templating. LESS (Less, n.d.) was used for preprocessing. Use of JavaScript also included its variants such as AngularJS (AngularJS, 2016) and jQuery (jQuery Foundation, 2015). MEAN stack was used as the framework and AWS (Amazon Web Services, 2016) was used for hosting.

4.1.3 The Strategy

The support tool is not out-and-out persuasive system but includes some persuasive features. The rating system (see section 4.2.1) is a visible persuasive feature implemented. It used direct approach to convey the opinion of the users to the contributor about the article. The concept of concealing the user's identity, while rating, makes the opinion more objective than revealing the user's identity.

4.2 Software Features

The features identified and implemented in the support tool is a result of brainstorming session on how to solve the problem domain i.e. lack of examples about PSD model. The initial brainstorming session during software feature identification phase included three individuals. The individuals involved were persuasive system experts and software development professionals. During the implementation phase, two experts from persuasive systems and software design field were introduced to provide feedback on the system and persuasive nature of the system.

The outcome of the brainstorming session was a concept to develop a system which supports people interested in designing and developing persuasive systems based on PSD model. Technically, it may not be a unique system in the sense that, there are other systems such as Stack Overflow (Stack Overflow, 2015), Quora (Quora, n.d.) that provide solutions and examples on different topics. However, it is an effort to solve the dilemma on the PSD model. Conceptually, it is a novel approach because no other system that we know of exists and attempts to explicitly provide categorized examples from persuasive design principles.

In terms of PSD model, the primary task of the system is to share the examples of individual principles of PSD model. It should provide users with categorized examples of each principle under respective categories. It attempts to lessen user's effort to find PSD model related examples from various sources by providing all of them under same domain. The obvious target users are, but not limited to novice users, researchers working on persuasive areas, BCSS designers and software professionals.

The support tool has used some principles from dialogue support category as basic guidelines. Attractiveness and ease to use have been central topics while discussing the system features. Praise and rewards have been used in rating feature which is a key component of the support tool.

System credibility category in PSD model includes principles that enable designers to create more credible systems. The support tool should look and feel credible. The users contributing must also be verified. Each contributor must have valid credential before they are allowed to upload examples to the site.

The support tool uses significant numbers of principles from Social Support category. The rating system and listing of articles utilizes competition and comparison while displaying. At the same time, there is cooperation because the users are marching towards common goal of strengthening the PSD field.

4.2.1 Rating System as a Feature

Rating system in the support tool is probably the most easily recognized persuasive feature implemented. There are two reasons for its significance. First, it allows both registered and unregistered users to rate the examples from different categories and design principles. Second, the examples categorized in each principle are listed and displayed to the user based on average rating. The average rating is the ratio of number of rating to the number of people who rate the particular example.

In terms of application of PSD principles, rating system utilizes Social Comparison, Competition and Recognition. When examples are displayed on the basis of rating, there is a feeling of comparison among the contributors. This comparison acts as motivating factor to be more active as it offers recognition. The Social Recognition persuades contributors to upload better examples, which would be featured in the list. This process augments sense of competition among the contributors. It is a positive influence as the contributors tend to compete with others and try to upload more relevant and real-world examples to get more recognition from the community.

4.2.2 System Features and Functionalities

Table 2 demonstrates correspondence of features and functionalities with persuasive design principles followed by description of each feature and functionalities in accordance to design principles.

Table 2. Features and functionalities and principle correlation

Features and functionalities	Design principles
Home page	Tunneling, Suggestion, Expertise
Category pages	Tailoring
Example pages	Tailoring, Praise, real-world feel, competition
Login page	Trustworthiness, Verifiability
Upload page	Cooperation
Profile page	Self-monitoring, recognition

Home page: The homepage is a first view that users see when they visit the application. This view provides basic overview of PSD model and its categories. It acts as initial point for navigation to other parts of the system. It allows users to proceed towards specific category or design principle of PSD model. This view also acts as navigation to an important Upload Example feature. The homepage also allows users to provide their feedback or comments regarding the support tool or examples uploaded. The system notifies the user if the feedback was successful or not.

Category pages: These are overview pages about individual categories. The design principles of PSD Model are categorized under it. Since the PSD model has twenty-

eight design principles under four categories, the support tool also has four categories with their related principles. It is tailored according to the PSD model so that the users have clear and consistent view on categories, design principles and examples.

Example pages: The example pages include real life examples contributed by the users categorized under specific design principle categories. It provides option for the readers to give their opinion on each of the articles, which affects the rating of the article. The examples are listed according to the rating system (see section 4.2.1 for rating system as a feature). This listing depends on user's rating contribution. The design principles such as praise and competition are clearly visible. The support tool uses Tailoring and Real-World feel to curate Real-World and related examples in same section.

Login page: As the name suggests, it acts as an entry point for the registered users who can perform activities such as contribute or see own contribution. If the users do not have valid credentials to log-in, they can register and get verified. The verified users can log-in to use the privileged features such as upload and user profiles. We could identify the principles such as Trustworthiness, Verifiability from System Credibility category in this functionality.

Upload page: In this page, the contributors upload the article. To be able to use this feature, users need to be registered, verified and logged-in. The contributor has options of providing the article with pictures and external URL. This is an act of cooperation in terms of PSD model where the aim is to share and obtain examples from PSD design principles.

Profile page: This page is for logged in users to see their contribution. The principles such as Self-Monitoring enables user to see their total contribution. Social -Recognition persuade the users to contribute more relevant examples and rate examples from other users.

4.3 Software Artifact and its development process

The identification of primary target users was easy to uncover. We tried to make the users contribute by implementing social recognition, one of the features in the PSD model. To have a successful software system, apart from identifying user, it is important to understand the user needs and documenting them for design and implementation (Nuseibeh & Easterbrook, 2000). The following section covers requirement study.

4.3.1 Software requirement

A common understanding among the stakeholders about requirements is a good start to a better software product. Sometimes it could be precarious due to unclear requirements or difference in priorities between the stakeholders (Karlsson & Ryan, 1997). The requirements for the PSD support tool were relatively easy and straightforward. The author of the artifact has attempted to make the use case easier and trivial without adding complexities. Figure 4, shown below describes the user case followed by the description.

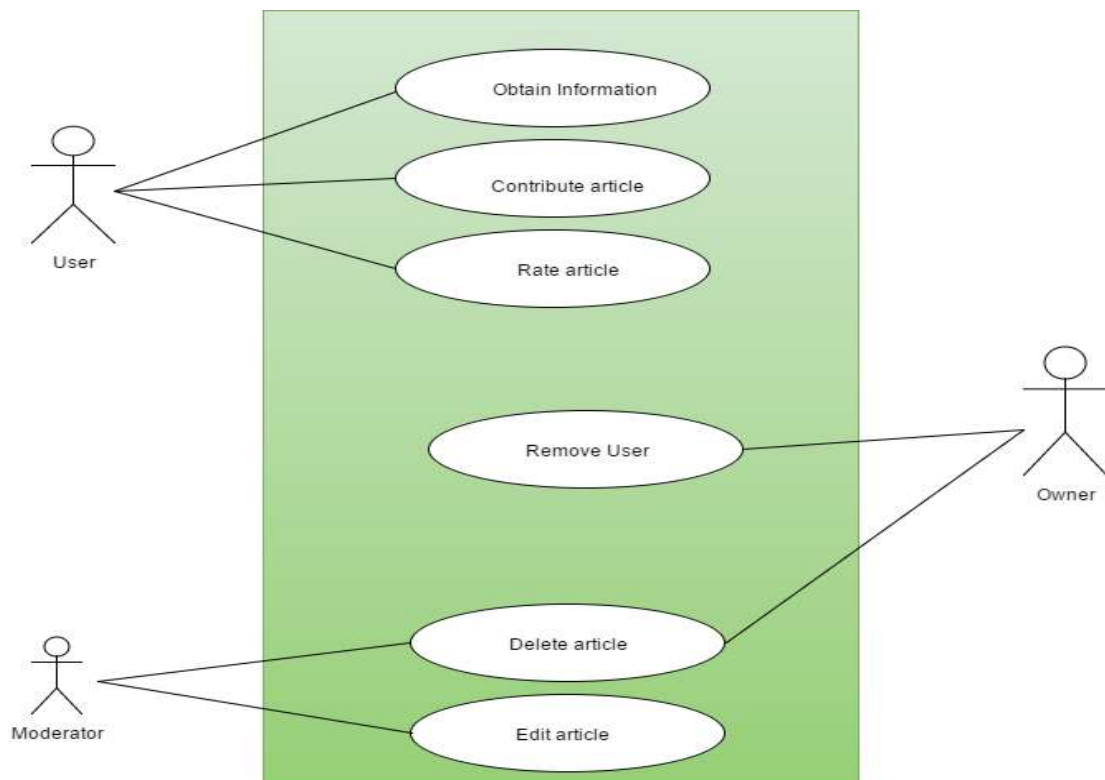


Figure 4. Use case diagram of the support tool

The use case diagram in figure 4 mentions a fairly simple use case. Users could perform three activities: obtain information, rate other's post and contribute themselves. For the later activity, users must be registered and logged in. The tool provided a user-friendly way to perform the registration and logging in. Users have right to add or rate the data but are not allowed to delete or edit the data until they are assigned as moderator. The moderator is allowed to add, rate and moderate other's post. The owner has right of user and moderator, in addition the owner has the privilege to remove user from the system if they behave inappropriately.

4.3.2 Software architecture

Software architecture is the representation of higher level abstraction from which software design and implementation are derived (Garlan & Shaw, 1993; Perry & Wolf, 1992). It enables designers to understand the structure, composition, interaction model as a whole system or separate entities. It acts as a bridge between requirements and implementation (Garlan, 2014). As the system gets complex, the application of correct architecture becomes essential for error-free representation of the software (Garlan & Shaw, 1993).

As shown in figure 5, a 3-tier architecture comprises of Client Tier, Business Tier and Data Tier. Client Tier is responsible for frontend processing and communicating through HTTP protocol. Business Tier comprises of application logic. Data Tier is responsible for data storage (Urgaonkar, Pacifici, Shenoy, Spreitzer & Tantawi, 2005). There are visible advantage of having the separate tier over one big chunk of tier such as reliability, security, usability, scalability, maintainability, availability, and time-to-market (Offutt, 2002).

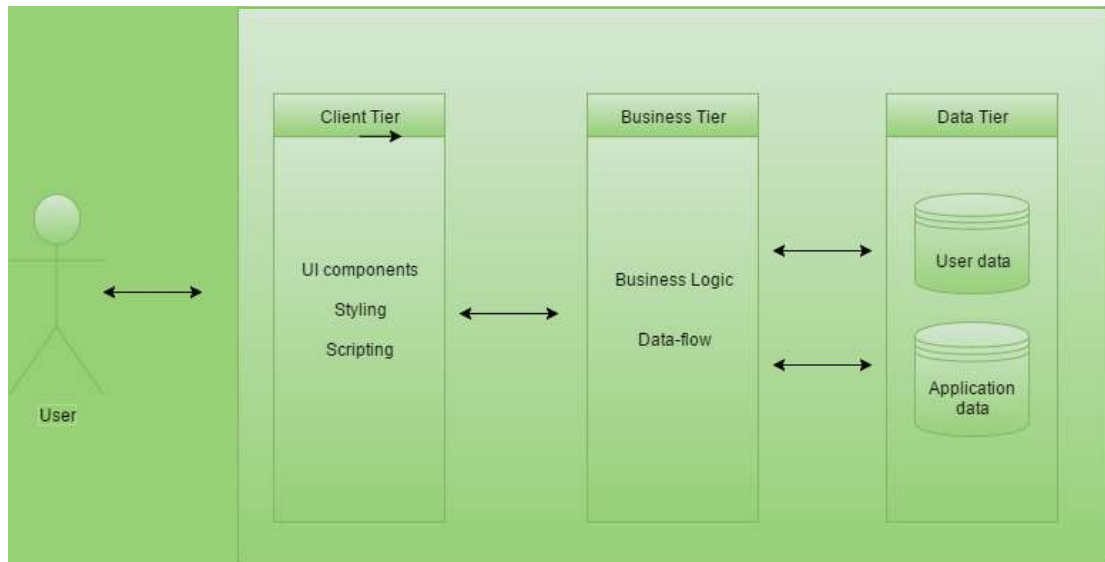


Figure 5. 3-Tier architecture of the support tool

Each tier stands on its own and is independent of other tier. If first tier needs to be improved or modified over time, the other two remain stable regardless of change in one tier. It might not be absolutely necessary in the smaller projects but as the system grows, the need for multi-tiered architecture becomes noticeable. The server layer and database layer and can communicate through middleware (Offutt, 2002).

In the front-end, AngularJS allows two way data-binding to synchronize the view with controller through dependency injection. The Model-View-Controller design pattern provides AngularJS with better structure, modular nature, better maintainability, and testability (Green & Seshadri, 2013).

Express, on the other hand, helps to design MVC-like structure for better structure and data interchange with database in the back-end (Mardan, 2014). Finally, the data tier communicates with the server layer's controllers to obtain and provide data. It does not activate by itself, instead it reacts to the request from the business layer. The support tool utilizes loosely coupled MVC in both front and back end. The reason behind this preference is the modularity we could gain, which will prove massive when the system starts to grow (Selfa, Carrillo & Boone, 2006)

4.3.3 Software Implementation

This chapter constitutes tools and techniques used during the implementation phase and the work-flow of the artifact.

Tools and techniques

HTML5 was used to markup along with Bootstrap for responsive front-end view. The worldwide consortium (W3C) specifies HTML5 as “fifth major revision of the Hypertext Markup Language (HTML), the format used to build Web pages and applications, and the cornerstone of the Open Web Platform” (W3C, 2014). Bootstrap is a front-end framework for creating mobile-first and responsive web applications. CSS3 was used for styling. JavaScript is the standout technology in the support tool because it was used to perform functionality from front end to backend.

AngularJS was used for scripting with occasional vanilla JavaScript and jQuery. jQuery is a lightweight JavaScript library used for DOM-manipulation, asynchronous data

exchange, animation and event handling. jQuery was necessary because many external libraries and modules used in the support tool such as Bootstrap depends on jQuery. Node.js was used as a platform to run express.js server. The reason for choosing Node.js, apart from being open source, is its power to scale and incredible performance (Node.js Foundation, 2015). The support tool uses AngularJS Full-Stack generator, which is basically a Yeoman generator, to scaffold MEAN stack boilerplate code (YEOMAN, n.d.). The Yeoman workflow, along with Yeoman, consists of Grunt as a build tool and Bower as front-end package manager.

Express.js was used as the web-server framework because of the middleware available to work with node and the possibility of using REST API (Express, 2015). MongoDB was used to store data (MongoDB, 2016). Mongoose was use for object modeling (LearnBoost, 2011).

In addition to the above mentioned tools and techniques, few other tools used are worth mentioning. The software was deployed in AWS Ubuntu 14.04 server, powered by Amazon. Compose.io was used for remote database, and Stormpath for user authentication (Compose, 2016; Stormpath, 2015). GitHub was used as remote repository host and putty.exe was used to transfer the code to the remote repository (GitHub, 2015; PuTTY, 2015). A handful amount of open source middleware such as connect, busboy were used to make the implementation faster and easier.

Implementation of Support Tool

In our case, use of Bower as front-end package manager could prove very useful when the amount of external packages used starts to grow. The node package manager (npm) makes easy to install and update packages for both front and backend shared by other developers (npm, 2016). For instance, the following line of code allows us to install the generator, Yeoman, Grunt, and Bower that we use to scaffold, run, and manage the application packages.

```
npm install -g yo grunt-cli bower generator-angular-fullstack
```

The result of running the generator is a project with client and server directories along with necessary configuration files. The client directory contains modularized front end components whereas the server directory contains the business logic and data exchange process.

Client Tier

In the support tool, front-end represents the client tier in 3-tier architecture. The modularization was done according to 28 design principles of PSD model, which means the client directory has 1 module for each design principle. In addition, it has modules for major functionalities such as login, upload, admin use, user profile, and factory. Each module has a HTML file for view, CSS file for stylesheet, controller file for declaring a controller for each view, JavaScript file for functionality Spec.js file for test. The factory is a reusable module used by all the modules as a model to get/supply data from/to the controller. It greatly reduces the lines of code and acts as glue in forming a MVC structure. The following code is an example of html code.

```
<div class="row row-small" ng-hover="background:red;"ng-repeat-
start="listin lists|orderBy:order:true" ng-click="showDetail(list)"
ng-class-odd="'alt'">

<divclass="rowrow-big"ng-show="active==list.title"ng-class-
odd="'alt'">
```

```

<div class="row">
    <div class="col-md-6 col-xs-12 row-big-font">
        <p> {{list.description}} </p><br/>
    </div>
    <div class="col-md-6 col-xs-12 row-big-font">
        <label>url:</label><ng-if="list.extUrl"><a
        href="{{list.extUrl}}"                target="_blank">
        {{list.extUrl}} </a></p>
        <p ng-if="!list.extUrl.length"> None </p>
    </div>
</div>
</div>
</div>

```

The above mentioned code illustrates markup, styling and angular directives leaving the unnecessary complicated logic from the html. It contains *list* which provides data to the template and is represented as `$scope.lists` in the controller. AngularJS binds the template with controller with its unique two way data binding such that change in either template or model reflects in other. Our support tool uses AngularJS factory to further modularize code so that we minimize the amount of logic in controller and separate model from controller. The following code snippet illustrates one such example.

```

var restModule = angular.module("restmodule", []);
restModule.factory("restfactory", ['$http', function($http) {
    var restFactory = {};
        restFactory.getData = function (urlBase) {
            return $http.get(urlBase);
        };
    return restFactory;
}]);

```

The above mentioned code snippet is an example for getting URL from controller and performing HTTP GET action and passing the data to the `$scope.lists`. It may seem redundant to use the factory for a single page but the support tool currently has more than 30 modules using this factory and will possibly increase as the system grows. The controller only needs to inject *restfactory* as dependency and will be able to use it to get the data from URL. The following code snippet illustrates its use.

```

angular.module('app') .config(function ($stateProvider) {
    $stateProvider .state('reduction', {
        url: '/reduction',
        templateUrl: 'app/reduction/reduction.html',
        controller: 'ReductionCtrl'
    })
}).controller('ReductionCtrl', function( $scope, $state, restfactory,
$stateParams ) {

```



```

restfactory.getData(url).success(function (data) {
    $scope.lists = data;
    }).error(function(error) {
        console.log(error);
    });
});

```

Business Tier

This layer is responsible for business logic and exists on the server (Microsoft, n.d.). In the support tool, Express separates the client tier from the data access layer. The support tool was deployed in Amazon Web Server which acts as a bridge to exchange data between client and data access layer. The client tier interacts with business tier through routing methods which are basically extension/derivation of HTTP methods. Routing methods are attached to instance of express middleware (StrongLoop, 2016.) The following code is an illustration from the implementation.

```

/* import express middleware */
var express = require('express');
/* Setup server */
var app = express();
app.get('/reductionData', function(req, res) {
    routes.getReductionData(req, res);
});
app.route('/*', function (req, res){
    res.render('homepage');
})

```

In this illustration, the express middleware was imported into the app through require ('express') method. This allows us to instantiate the express into app variable and subsequently call GET method to get data from reduction design principle. The client tier only interacts through defined endpoints to retrieve from or provide data to data access layer. If the client attempts to use undefined endpoints it is redirected to the homepage. The support tool uses Stormpath middleware for user authentication and API protection. For instance, adding authentication middleware in above mentioned method to get reduction data results into:

```

app.get('/reductionData', stormpath.loginrequired, function(req, res)
{
    route.model.getReductionData(req, res);
});

```

Adding Stormpath middleware requires the users to be logged in before they can get the data regarding the Reduction principle. This layer does not fetch data from the data by itself. Instead, it asks data access layer to query database and provide the data.

Data tier

This data tier interacts with external database to retrieve and store persistent data according to the business layer's instruction. Having a separate data tier layer allows to manage and maintain the system without interrupting other layers. The components in data tier are modularized and serve a definite purpose thus making it more reusable than client-server architecture (Microsoft, n.d). The support tool uses Compose.io to host MongoDB for better scalability and maintainability. Mongoose was used to write schema based query. The schemas are used to create models. The models are instantiated to create documents which are stored on the database. The following code snippet provides is an example:

```

/* Create a new Schema using mongoose*/
var reductionSchema = new Schema({
  idProduct: {
    type: mongoose.Schema.Types.ObjectId,
    index: true
  },
  title: {type: String},
  description: {type: String},
  username: {type:String},
})
/*Use the Schema to create a Model*/
ReductionModel = mongoose.model('reduction', reductionSchema );
/*Create an instance and save it in database*/
var reductionExample = new ReductionModel({title: 'example' ,
description: 'description of example', username: 'user1'});
reductionExample.save( function (err){
  if(err) {
    console.log('Unable to process your request');
    res.render('creationPage');
  } else {
    res.render('homepage');
  }
});

```

The above code snippet is an illustration of creating Mongoose schema and converting it into a mongoose model. The mongoose model is then instantiated to create an example of Reduction. It checks for validation and renders home page on success and redirects back to creation page on failure.

Software workflow

This section demonstrates the detailed workflow of the support tool. It comprises of activity diagram creates with the aid of UML and its description. The UML diagram is

used to depict the higher level representation in graphical format using action, relation and decision without over-complicating the scene and keeping it simple (Tilley & Huang, 2003). The support tool includes 3 major activities: browse, rate and contribute. Figures 6, Figure 7, and Figure 8 illustrate the activity diagrams for browsing, upload, and rating.

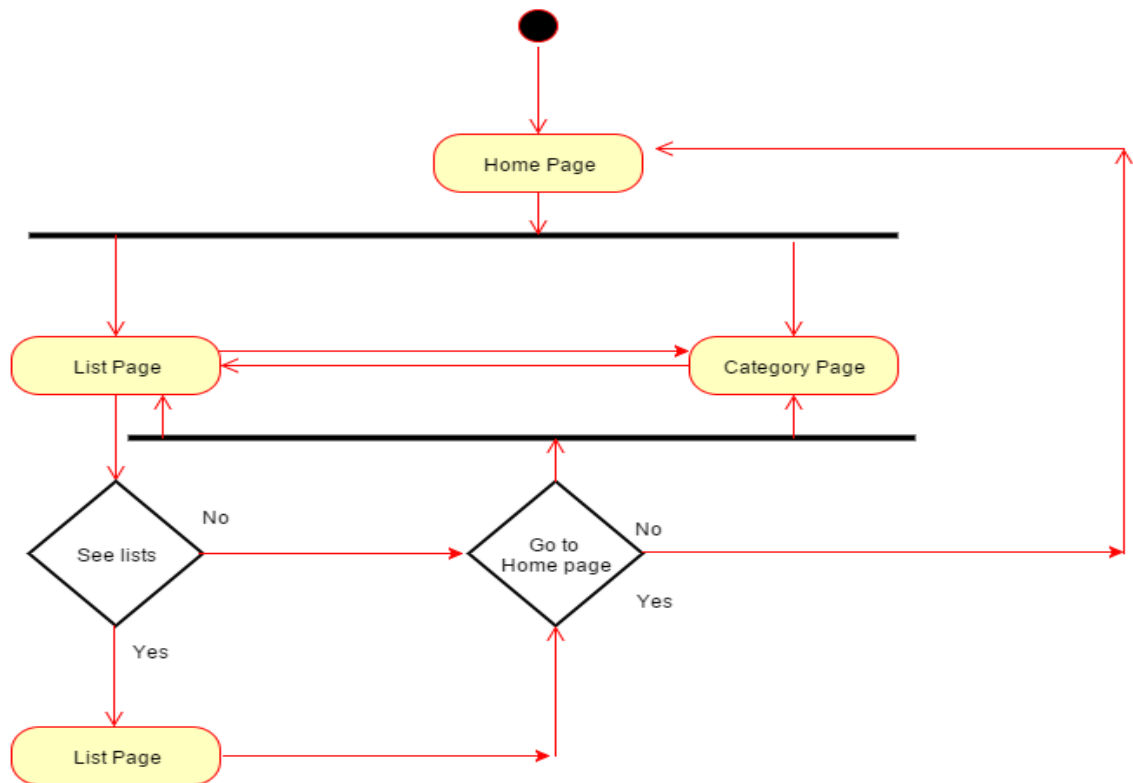


Figure 6. Activity diagram of browsing feature

Figure 6, shown above is a visual representation of how browsing feature should work. The activity usually starts from the homepage. From the homepage, users have options of navigating to category page or list page. Category pages and list page have navigation bars so users can browse to desired sections. Since category pages contain information about different principles of PSD model, each principle is linked with its list of examples. The list of examples for each principle is displayed in its own list page. The examples contain details and rating feature (see figure 8 for rating activity and its description). The list pages also have navigation functionality to go to other sections of the system including the homepage. Upload is one of the important functionalities of this system.

Figure 7 demonstrates how the user can use the upload feature. For better user experience, the user should be able to navigate to the upload page from every section of the application. In order to be able to upload, the user should be logged in. If the user is not logged in, the application will itself navigate the user to the login page. For the user to be able to contribute, they have to be registered and logged in. In order to register, the user needs to fill the registration form and be verified using the email they receive. After the registration process, users are eligible to upload. A user can login or opt out from the upload process anytime. If users wish to upload an article, they can select sub-topics and upload with or without image.

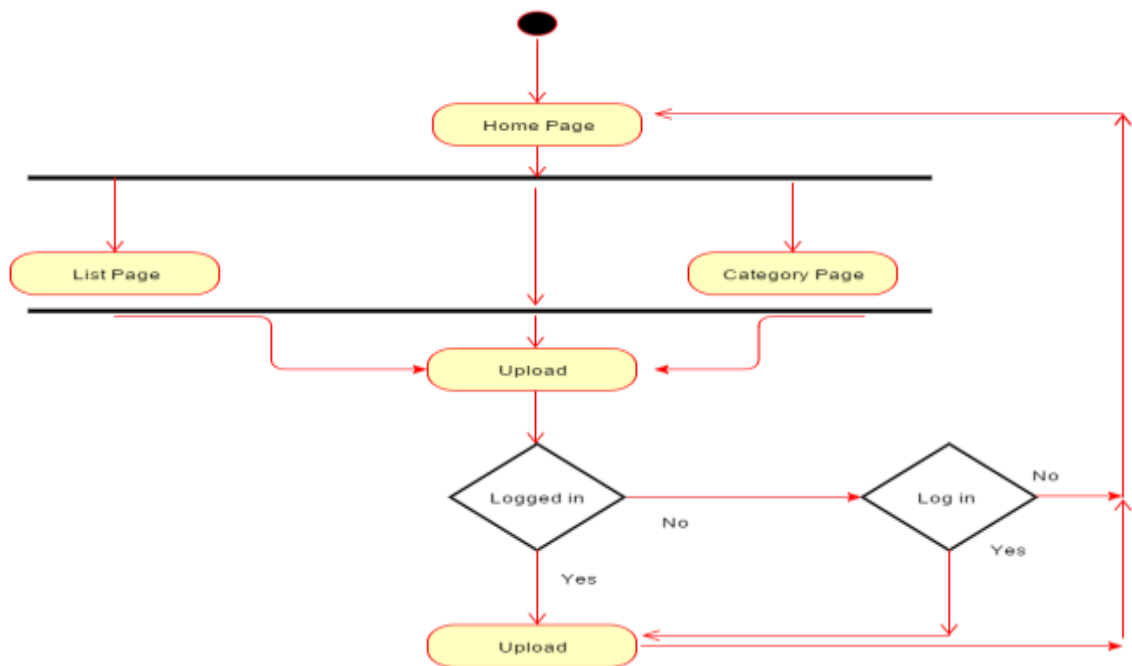


Figure 7. Activity diagram of upload feature

Figure 8 illustrates rating activity. If the user wishes to rate any pre-existing example, they can do so by activating rate workflow. The user has the option of proceeding with rating the example or discards the rate activity. If the user wishes to discard the rating activity, then the user has option to continue browsing or starts contribute activity.

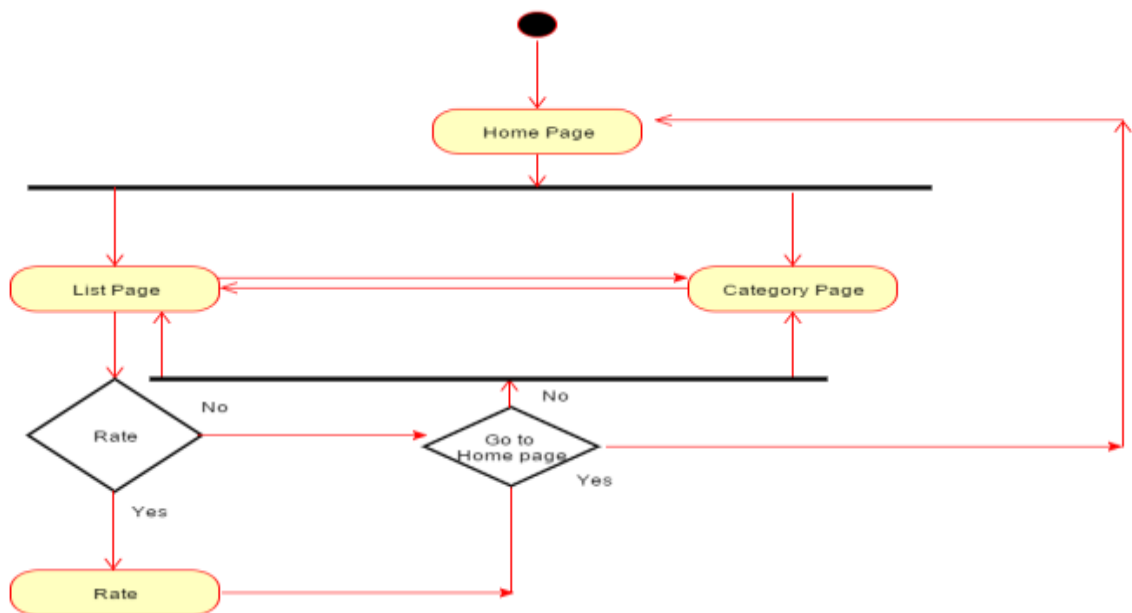


Figure 8. Activity diagram of rating feature

4.4 Artifact Demonstration

This section demonstrates the artifact produced. It comprises of screenshots of major parts of application along with their brief description about the structure, functionality and navigation.

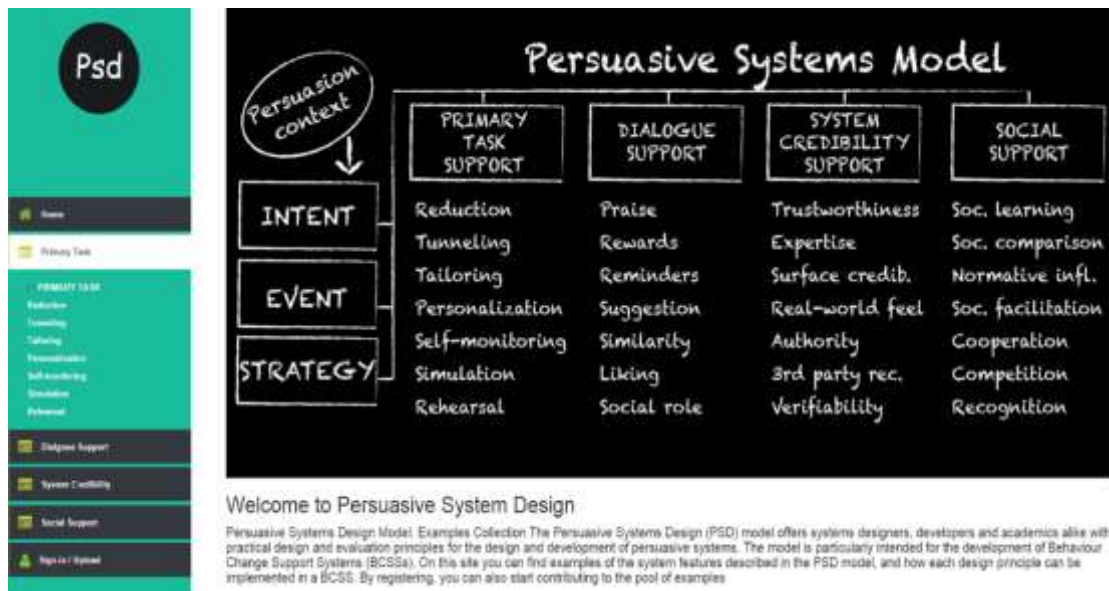


Figure 9. Screenshot of Home page

The figure: 9, shown above is a screenshot of landing page. The landing page consists of a link to home page itself, different categories of PSD model (refer to section 2.2.1) and link to contribute section. It consists of responsive sidebar for navigation and main content page to let users know more about the PSD model, about the team and provides the response section where users can send their message to the admin regarding the support tool. The next view is listing of examples.

The screenshot shows a list of persuasive system design examples, sorted by rating. Each row includes a thumbnail, a title, a rating (stars), and a plus sign for more details.

Thumbnail	Title	Rating	Action
[Thumbnail]	Show-off your achievements	10 ★	+
[Thumbnail]	Polar Flow	9.67 ★	+
[Thumbnail]	Your organization's overall recognition culture	8.4 ★	+
[Thumbnail]	Innovative doctor saves life	4 ★	+
[Thumbnail]	Fitbit	0 ★	+
[Thumbnail]	REC	0 ★	+
[Thumbnail]	Michaelbauer	0 ★	+
[Thumbnail]	Suunto Movescount	0 ★	+
[Thumbnail]	Student	0 ★	+

Figure 10. Screenshot of List detail view

The list view is shown in figure 10. It displays the examples of persuasive system design principle in the order of rating. The example with highest rating is displayed at the top and it continues in decreasing order. Listing of article in this way allows users to see the most rated examples first. It also entices the contributors to upload more relevant examples. The next view is detail of this list view.

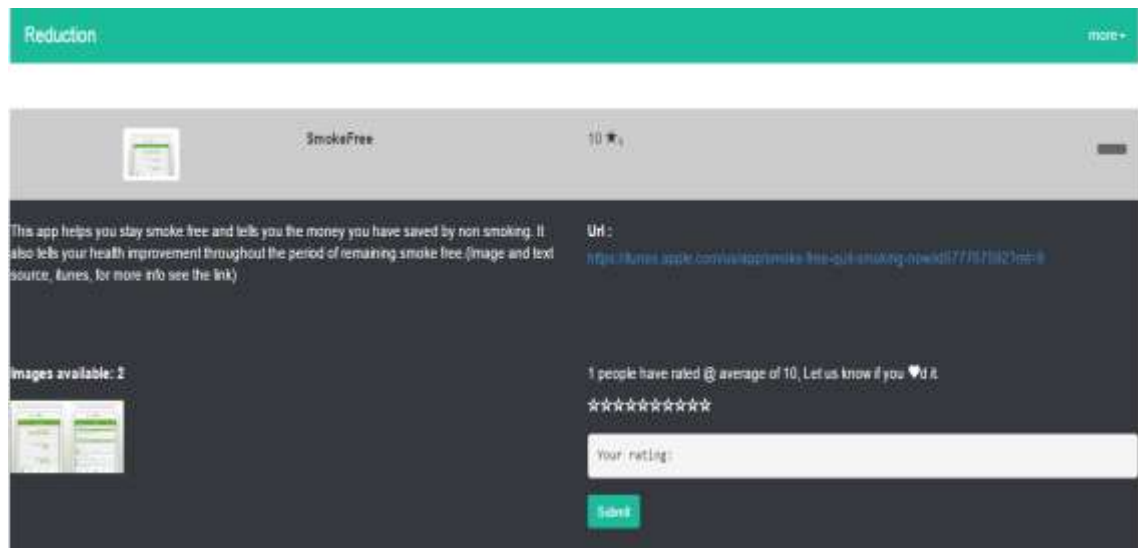


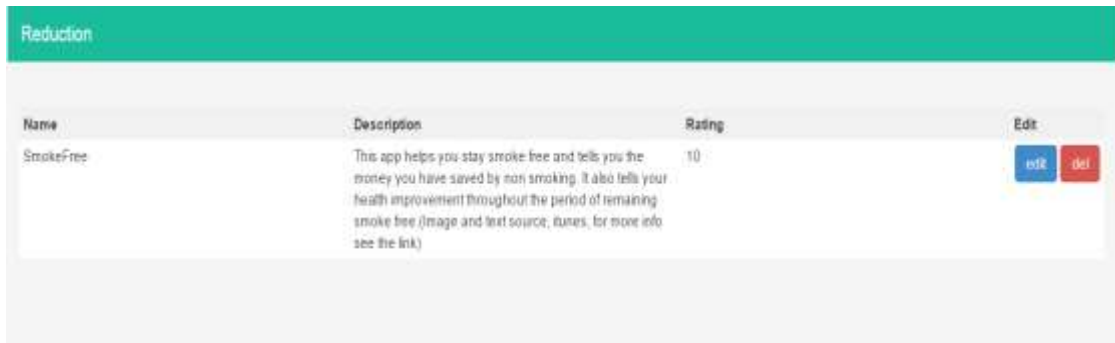
Figure 11. Screenshot of List detail view

By clicking the listed examples, users have option to rate the article if they wish to. To avoid any complication the system does not prompt any questions rather it updates the rating, disables the rating button, and moves the rated article to the top. The listing is done in real time so that the users could see the effect of their rating. The next view is upload view.

 A screenshot of a web form titled 'Upload Form'. The form has a light blue header. Below the header, there are several input fields: a 'Username :' field, a 'title' field with a person icon, a large text area for 'Please describe your contribution...', and a text field for 'Please provide Uri if any...'. Below these is a dropdown menu labeled 'Select the PSD model : Choose...'. Underneath is a section for file upload with a 'Choose Files' button and the text 'No file chosen'. At the bottom, there is a 'submit' button and the text 'Are you ready to proceed ?'.

Figure 12. Screenshot of upload page

Shown in Figure 12, upload view allows verified users to contribute examples of persuasive system design principles. Unlike rate and other PSD article example pages, the user needs to be authenticated to be able to contribute. The username is extracted automatically when the user logs-in. Title and description and selection of PSD models are mandatory fields to be provided for the user to contribute. The fields such as external URL and image are not mandatory. The next view is admin view.



Name	Description	Rating	Edit
SmokeFree	This app helps you stay smoke free and tells you the money you have saved by non smoking. It also tells your health improvement throughout the period of remaining smoke free (Image and text source, images, for more info see the link)	10	edit del

Figure 13. Screenshot of admin page

Figure 13 illustrates admin view which is not available to normal users. This page is only available to authorized personnel. In order to operate any admin activities the personnel must be logged in. With the admin view, the admin could edit the article or delete if it violates the terms of use. However, the admin is not able to rate the article so that there is no admin bias.

5. Evaluation

Along with a good architecture, assessment of functional and nonfunctional requirement is vital for success of any software system (Williams and Smith, 1998). In order to understand the current status and improve the system, the software should be evaluated accurately (Casal, Tubío, Vázquez, Fernández, & Yáñez, 1998). According to Hevner et al., (2004), “The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods”. The iterative nature of design also requires rigorous, reliable and appropriate evaluation methodologies to further improve the quality of any software artifacts (Hevner et al., 2004.)

Evaluation must be rigorously conducted if the designed artifacts are to serve its purpose. Without evaluation, the study cannot prove that designed artifacts can a) solve an existing problem b) improve some existing system (Venable, Pries-Heje, & Baskerville, 2012). We conducted an experiment to understand and analyze whether the app was able to serve its ambition of being a support tool to aid in designing persuasive systems. Persuasive systems are not yet ubiquitous. However with the emergence of BCSS, the future looks promising. We believe that, for designers and software developers to adopt our support tool, it should provide better experience and match their needs. Furthermore, we wanted to understand whether the support tool was desirable and would provide prolonged solution rather than short-term aid. The following section describes research setting and explains the nature of experiment conducted.

5.1 Research Setting

This section provides insight about objectives, participants and methods applied during the evaluation process.

5.1.1 Objective of evaluation

The primary objective of this evaluation was to understand whether the support tool has achieved its goal on providing categorized examples of different design principles of PSD model. We derived a set of task to be performed mentioned in Table 3, to understand if the app served its fundamental purpose. Software development is a cycle of development, evaluation and improvements (Hevner et al., 2004).

Our understanding is that the tasks stated in Table 3 serves basic purpose of app’s existence. With the evaluation results, we plan to improve the tool to assist wider audience in future upgrades. The next section explains the background and areas of expertise about the respondents who participated in the evaluation process.

Table 3. Activities to perform during evaluation

Task no.	Task description
1	Browse through different categories and examples of design principles.
2	From the system, find examples of social recognition from social support category.
3	When you find the example of social recognition, click on the example and see if you can see the detail view. From the rating bar in the detail view, give the example some rating based on whether you feel the example you clicked is relevant to social recognition.
4	Navigate to Sign-in / Upload page and click register tab in the top of the navigation bar. Fill the form to register, it will send you a verification link to the email address that you provided. Verify that the registration is complete by clicking on the link in your email.
5	To Upload an example, navigate to Sign-in / Upload page and select upload tab from navigation bar. Fill the title and description, select the recognition from select bar and upload at least two relevant picture of social recognition.
6	See if someone else has rated your example.

5.1.2 Evaluation Participants

The evaluation process involved experts of persuasive field as users. To start with the evaluation, we invited 8 persuasive experts and received 5 responses. The response rate is over 62%. Out of 5 respondents, who participated in the evaluation, 3 were females and 2 were males. The demographic structure according to gender is 60% female and 40% male.

We distributed the respondent's expertise into 0-3 years, 3-6 years, and 6 or more years. 2 respondents had less than 3 years of experience, 1 had 3-6 years of experience and 2 had more than 6 years of experience in persuasive technology field. The subdivision enables us to understand expertise level of evaluators. The response suggests that the expertise level of respondents varied and also did not represent opinion from single expertise level. In addition to expertise in persuasive systems, 2 of the respondents have experience in usability, HCI, software design/development. Having experts in multiple field increases the probability of finding usability problems and increasing performance of the system (Nielsen, 1994, p. 161).

For better understanding about the participants, we asked whether they themselves are users of persuasive technologies. The result was mostly affirmative. Out of 5, 3 respondents mentioned they use persuasive technologies, 1 said sometimes and 1 said no. From the response, we could deduce that majority of respondents themselves use persuasive systems. This information makes our evaluation more reliable. The evaluators were asked to perform activities in Table 3 that seem imperative for the existence of the system. They were allowed to use the artifact freely on their time of

convenience. The normal time to respond was 4 days and later extended to additional 3 days to include more respondents seeing their occupied schedule.

5.1.3 Evaluation method

The term *Usability* is defined as “The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use” (as cited in Jokela, Iivari, Matero, & Karukka, 2003). Along with the identification of a) users b) user goals and c) context of use, the measurement of effectiveness, efficiency and satisfaction should be determined in order to fully understand the usability of any product (Jokela et al., 2003).

Barnum and Palmer (2010) mention that effectiveness and efficiency of a system can be determined by conducting usability evaluation. However, using usability evaluation might not be as effective in uncovering satisfaction. This method has not been particularly successful in exploring intangible aspects like fun, enjoyment or desirability (Benedek & Miner, 2002). We agree with Barnum and Palmer (2010) on the reasoning that user’s satisfaction may have crucial impact on whether user uses the system or not. Satisfaction of user should not thus be neglected. Rather it should be welcomed as an opportunity to understand user’s feeling about the system (Barnum & Palmer, 2010).

To test effectiveness and efficiency of activities from Table 3, usability evaluation could be better a choice as it uncovers these two attributes of usability testing. Still, it leaves out user satisfaction, the third and vital aspect of usability (Barnum and Palmer, 2010). Semi-Structured Usability Evaluation was conducted during development phase to gather expert’s opinion about the support tool.

After gathering usability feedback from the experts we wanted to understand user’s feeling about the system. The idea was to understand a) user’s satisfaction b) did they perceive the rating system as effective and c) their intention to use. To understand user’s satisfaction we decided to test desirability of the support tool. The desirability test uses Product Reaction Cards which is a part of Desirability Toolkit (Benedek & Miner, 2002). We also opted to evaluate persuasive effectiveness of the rating system and support tool to understand the perceived persuasive effectiveness and use continuance using constructs from Lehto and Oinas-Kukkonen (2015).

The evaluation comprises of three sections: a) Usability Testing b) Desirability test and c) Persuasive effectiveness. Usability testing was carried out as semi-structured expert evaluations during the development phase, where the findings contributed to development iterations. Desirability test and persuasive effectiveness evaluation were conducted after the development was complete. The evaluation was carried out using Google Form. It is free software from Google, Inc. and allows customization. The support tool evaluation required the respondents to login to Google form so that they could only submit their answer once. The evaluation form was distributed through email. Following sections illustrate the evaluation conducted in detail.

Semi-Structured Usability evaluation

The design and development of the support tool had two rounds of informal semi-structured evaluation. The process involved four individuals out of which two were persuasive experts, one software development professional and one was support tool developer who acted as facilitator. Each round was a semi-structured evaluation where evaluators took part in a specially arranged session and the facilitator guided them

through the sections of the system. But there were no specific pre-set questions that they all formally answered. Rather it was carried out as informal walkthroughs. In this process, the experts considered the system and its key use cases, pointing out potential issues and ways of solving them. During these rounds, primary functionalities of the tool were tested. Various issues and problems were discovered and solutions were also proposed by evaluators. Following section uncovers the major issues discovered and their solutions.

Issues and Solutions

Issue 1: Front Page

The evaluators discovered few issues in the home page of the tool. Missing links, consistency, and unnamed feedback forms were pinpointed during the evaluation.

Solution: To resolve the issues mentioned by the evaluators, the missing and broken links were fixed. The color combination used throughout the page was limited and made consistent. The feedback form was named to “Feedback Form” so that it clearly notifies the user about its functionality.

Issue 2: Navigation

Evaluators identified some issues regarding navigation. The evaluators wanted to have clear separation between categories and principles of PSD model. Technical issues such as inability to navigate from the logo and navigation to sub categories were also discussed.

Solution: A symbol is added to separate categories and principles. The navigation issue from log and other sub categories has also been resolved by adding links to respective pages.

Issue 3: Listing Page

The evaluators suggested that, subcategories and principles from PSD model should be defined. The system should provide evidence that they actually work. The evaluators also mentioned that, there should be ways to implement these principles.

Solution: The principles and categories have been defined in each category pages. Providing evidence and ways to implement each principle is currently out of scope of this study. However with the examples collected, it may be possible to implement these suggestions in future. It certainly opens a new paradigm for the support tool to not only limit itself in collecting categorized but also be persuasive on using them.

Issue 4: Registration / Login

The evaluators reported some issues during Registration / Login. The names and icons used should be changed to more relevant for example, “*Upload*” to “*Register to add example*”. Users should be provided with information on “what they can do after logging”. Only immediately available features should be displayed.

Solution: The icons have been changed and efforts have been made to name the functionalities to more relevant ones. We found comments a) to let users know what they can do after logging and b) display only immediate features somewhat

contradictory. Thus we opted to display features possible after logging-in. This decision may be altered depending on the response from desirability test and user's response.

To summarize, we received plenty of issues and their proposed solution in various section of the support tool. The evaluators have done incredible work in being so vigilant and meticulous. We have attempted to resolve the possible issues during development phases. There were some issues that require more research and some of them were out of scope from our current study. On the bright side, these responses will provide us guidance for our future upgrades.

Desirability test

The Desirability Toolkit (Benedek & Miner, 2002) is a result of brainstorming session at Microsoft to seek methods to understand abstract and intangible concepts such as fun. It consists of two methods to explore such aspects i.e. Faces questionnaire and Product Reaction Cards.

In the faces questionnaire, the respondents provide feedback about their emotion after using the product by looking on the picture of a face and rating accordingly. For example, if a user felt sad after using a product and sees a sad face she/he would rate 7 in 1-7 scale which means the picture reflects how she/he feels after they use the product. Product reaction cards allows user to select few, usually 5, words from a large pool of words to describe their feeling after using the product. The set of pool should contain at least 40% of negative words to negate positive bias towards the product (Benedek & Miner, 2002.)

According to Barnum and Palmer (2010), Production Reaction Cards from Microsoft reveal more information about satisfaction than other available techniques. The reason behind this leverage is, Product Reaction Cards encourages users to choose words that entirely express their feeling about the product (Barnum & Palmer, 2010).

To evaluate desirability, we used 118 adjectives from the Product Reaction Cards (see Appendix A) and asked the evaluators to select 5 adjectives that best represent the support tool. Along with selection the evaluators were asked to provide insight on why they selected those particular adjectives. The later was an open-ended question where the respondents were expected to provide their own view in creative way rather than limiting them to select from predefined list. The respondents were provided with an example on how to provide answers to avoid inconsistency and ambiguity. The results and findings from desirability test are listed in section 5.2.

Persuasive effectiveness evaluation

In this section, we construe the evaluation of perceived persuasive effectiveness of rating feature in the support tool. According to Lehto and Oinas-Kukkonen (2015), any system is considered resourceful if users use it to gain proffered benefits. Otherwise it remains mediocre despite having potential. Lehto and Oinas-Kukkonen (2015) further mention "perceived effectiveness has significant impact on continuance". Barnum and Palmer (2010) have also acknowledged effectiveness as key aspect of usability. Drozd, Lehto, & Oinas-Kukkonen (2012) in their research have demonstrated that intention to use have positive impact on actual usage.

During this part of evaluation we attempted to evaluate perceived effectiveness of the persuasive feature i.e. rating feature implemented in the support tool. By understanding

the perceived effectiveness we tried to understand the use continuance and actual usage of both rating system and our support tool. For this study, we used constructs based on Lehto and Oinas-Kukkonen (2015) and similar to Venkatesh, Thong, & Xu's (2003) performance expectancy construct. The appendix 3 illustrates the constructs as evaluation form and the results obtained are reflected in results section 5.2.2.

5.2 Results

This section illustrates responses obtained from the evaluators and attempts to synthesize the findings into a more readable result. Section 5.2.1 reveals responses from desirability test and 5.2.2 expound data from persuasive effectiveness of the social support feature of app i.e. rating feature.

5.2.1 Results from desirability test

To our wonder, evaluators have selected a wide range of adjectives to describe the support tool. Only 2 adjectives that were repeated are Simplistic and Usable. Figure 14 illustrates adjectives that were selected by the evaluators.

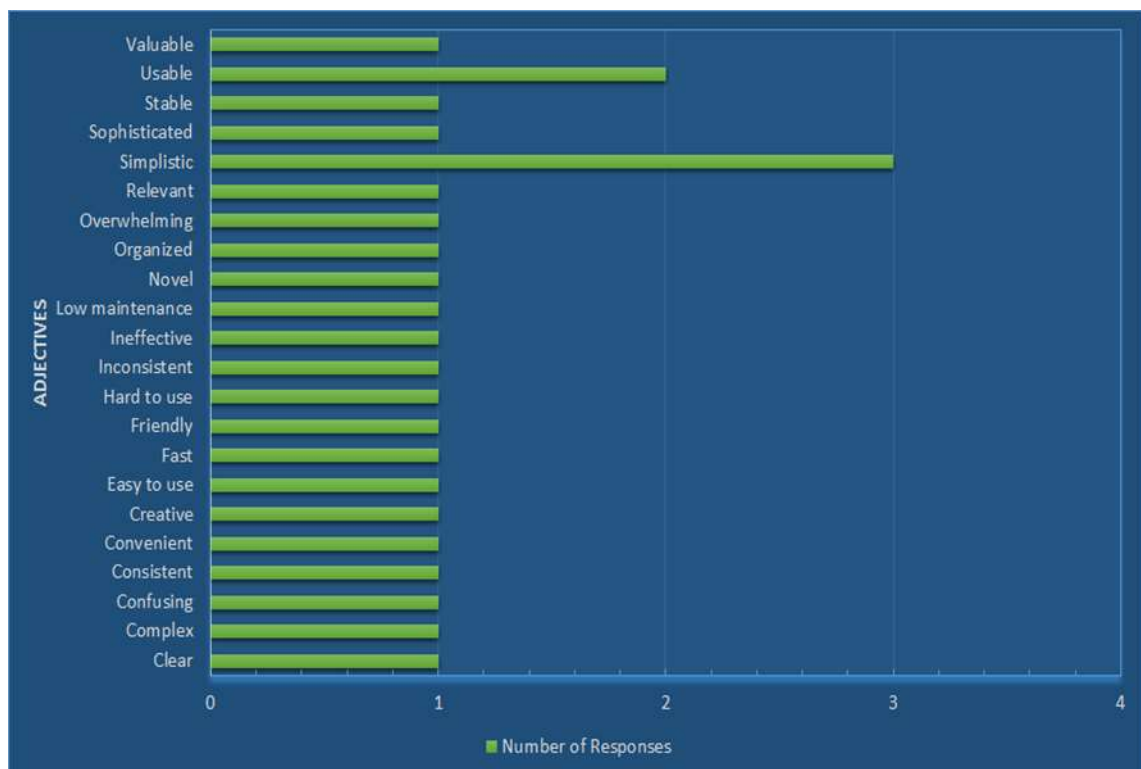


Figure 14. Adjectives selected to describe the system

This wide range of selection as illustrated in Figure 14 is a challenge to generalize but also an opportunity because it presents a diverse view towards the system which can be invaluable in upgrading the system. Out of 5 respondents, 3 felt that the design and layouts are simple and 2 feel it usable. There were positive feedbacks such as Easy to Use, Friendly, Fast, Stable, Organized, Convenient, and Consistent. Also there were negative feedbacks such as Inconsistent, Ineffective, Confusing, Hard to use, and Overwhelming. The respondents used adjectives such as Creative, Valuable, Novel, and Relevant to describe the support tool to describe novelty in persuasive system design area. The major contradictory issue was navigation where some people thought it was

Organized, Clear, and Convenient. At the same time others felt it was Hard to use, Inconsistent, and Ineffective. Table 4 represents few interesting comments.

Table 4 also shows that, evaluators have been active and open to provide the reason behind the selection of adjectives. The desirability has been a success in the sense that, respondents have provided their opinion freely with positive and negative adjectives nullifying the positive bias issue.

Table 4. Adjectives selected and their description

Adjectives	Description
Confusing	The navigation system wasn't very clear. E.g. when in Recognition, the home button was hidden.
Valuable	The content will create value.
Novel	Not many (if any) similar tools available
Complex	It might be difficult to use understand for those who are not familiar with the PSD model
Inconsistent	The main menu is accessible only via back stepping or very unintuitive item more. Menu button should be visible always (to access home at least, maybe also to access the main menu view). Inconsistent referring to layouts, they are different and do not follow the same style
Overwhelming (not the best word but almost)	When viewing the uploaded examples e.g. in recognition, the list items are colored as white and brown. This is unnecessary and has no real use. It helps to see the lines but anyway, the line itself is clickable and you do not need to press the + sign to open it. Coloring adds unnecessary distraction to the view.
Easy to use	It was relatively easy to use

5.2.2 Results from Persuasive effectiveness evaluation

The support tool displays articles uploaded on the basis of average rating. During persuasive effectiveness evaluation, evaluators were asked whether listing the articles on the order of rating improves the persuasive effectiveness of rating system, Figure 15 illustrates the response.



Figure 15. Listing of article based on order of their rating

Figure 15 shows that 1 respondent disagreed with the idea that listing of article in the order of rating improves persuasive effectiveness of rating system. 2 respondents remained neutral. Remaining 2 respondents agreed to that statement. Another question in persuasive effectiveness evaluation was, “Possibility of rating articles improves the persuasive effectiveness of the app”. Figure 16 illustrates the results from this question.

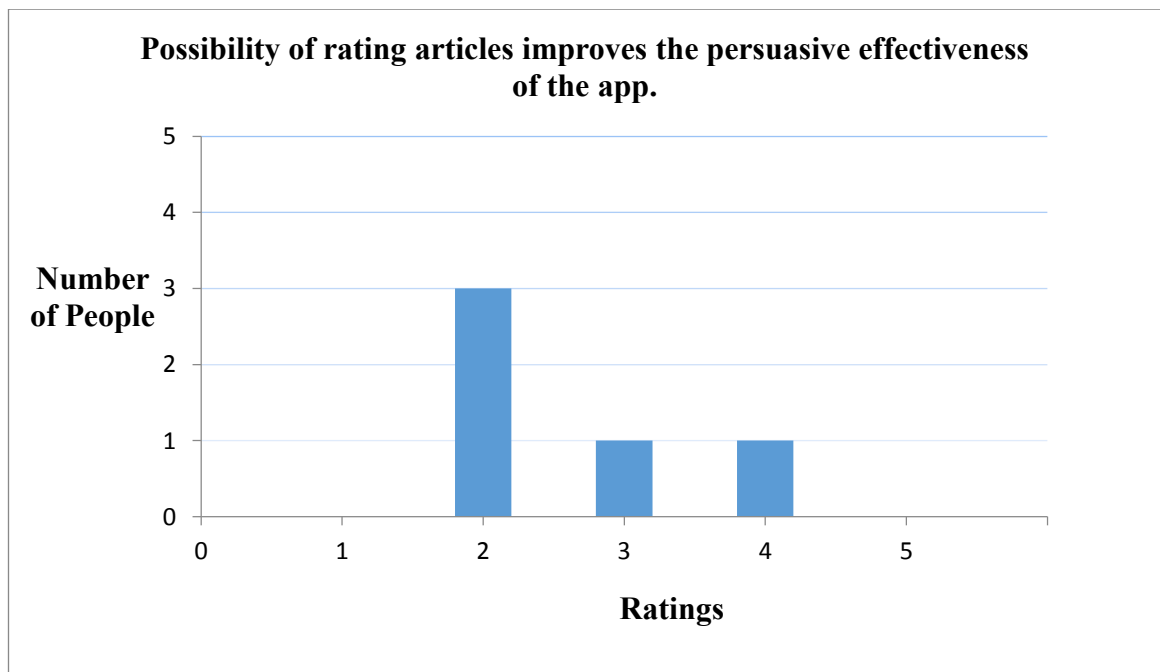


Figure 16. Possibility of rating articles improves the persuasive effectiveness of the app

Majority, 3 respondents disagreed to the statement that possibility to rate articles uploaded in the system improves the persuasive effectiveness of the app. 1 of the respondents remained neutral to this statement and 1 agreed to this statement. Another statement was whether, “Having a rating system makes the app nice to use”. The result of this statement is displayed in Figure 17 and explained in following section.

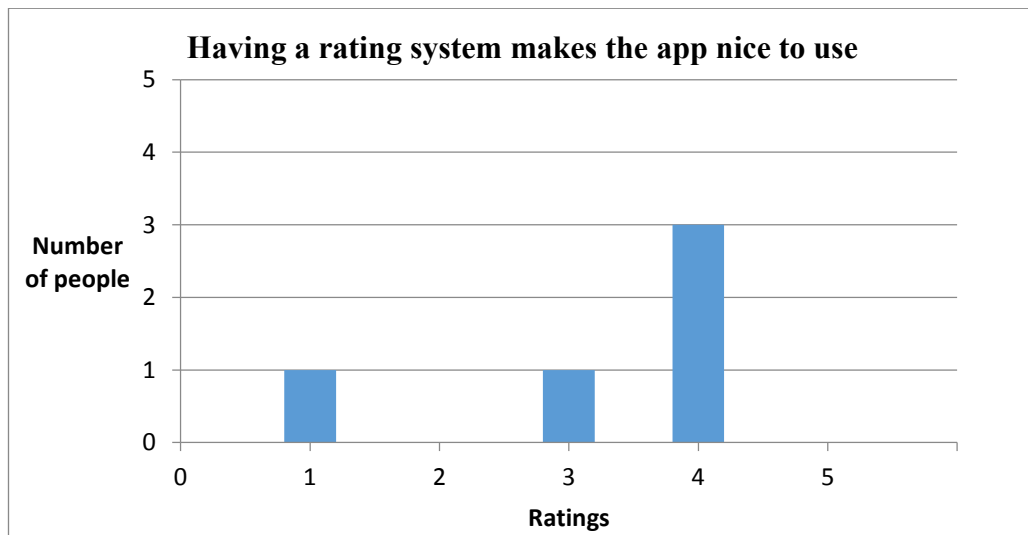


Figure 17. Having a rating system makes the app nice to use

1 of the respondent strongly disagreed to the statement that, having rating feature in the support tool makes the tool nice to use. 1 of the respondents stayed neutral on this statement whereas 3 respondents agreed to the statement. The final question about persuasive effectiveness evaluation was whether, “Having a rating system is important to the app”. Figure 18 is visual representation of response followed by the explanation.

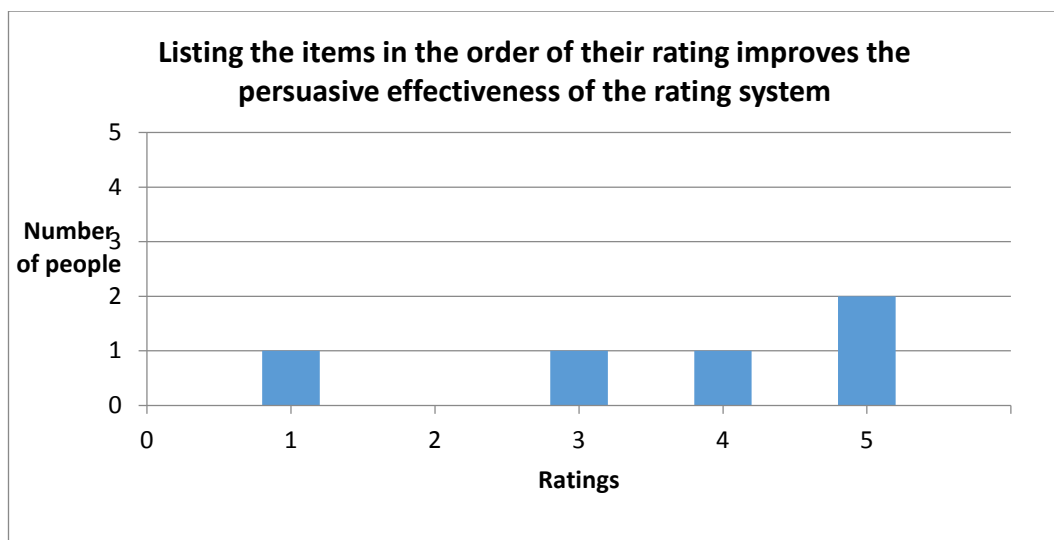


Figure 18. Having a rating system is important to the app

This statement gathered some interesting response. 1 of the respondents strongly disagreed to this statement, 1 remained neutral. 1 agreed to the statement and 2 strongly agreed to the statement that, rating system is important to the app.

5.3 Findings

It was an interesting mix of responses from the evaluators. Although we received wide range of responses, we could make a general assessment from the responses.

During the usability evaluation, the evaluators discovered quite a handful amount of usability issues. Along with those issues, they provided us with some insightful ideas to solve those issues as well. Having experts from persuasive, HCI, Software design and

development domain helped us to gather divergent issues. The respondents brought issues from UI design, navigation, missing functionalities, being more consistent with persuasive design principles, providing guidance to use complex functionalities and usability as a whole. The issues and the proposed solutions were carefully analyzed and necessary and possible changes were made during development phase.

Regarding the desirability evaluation, the respondents were mostly positive about simplistic, organized design, and usability. Some of the evaluators felt that structure was clear, friendly and consistent. Some mentioned about robustness in terms of stability and low maintenance whereas some said the system was easy to use. From the responses it was evident that some of the respondents were not happy how the navigation worked. They did not withhold their opinion to mention the weaknesses such as overuse of styles in some parts of the app. It was great to see that evaluators were being watchful and strict. An instance of such detailed observation was that registration process being complex and needing guidance. One of the respondents mentioned:

Hard to use: e.g. when registering, there wasn't instructions on the password, I needed several tries to get it right.

We acknowledge that some of the problems may have been overlooked and need to be addressed to make the system more user friendly and provide better user experience. There were some repetition in usability evaluation and desirability test such as navigation issues and needing guidance. We acknowledge that performing complex task need some guidance and we will make those improvements during the first public release. We also accept that navigation issues are of high essence in any app and it needs to be consistent. We will try to remain more consistent throughout the app and attempt to provide simple and easy to use experience to our users.

Admitting to the shortcomings the tool has, it is also understood that this support tool is necessary to persuasive system design domain. Most of the evaluators confirmed that this support tool brought novelty in its field and may prove valuable with improvements. Their experience with persuasive systems has helped us a great deal to understand the persuasive effectiveness of the app. According to Lehto and Oinas-Kukkonen (2015), at the center of influencing users are persuasive features. If the system cannot engage the users for longer period or retain them then all the improvements become obscure and diminish. From Lehto and Oinas-Kukkonen (2015), we understand that perceived effectiveness have significant positive impact on use continuance intention. Thus we attempt to assess the use continuance using the questionnaire on perceived effectiveness and aforementioned theoretical model.

To summarize the persuasive effectiveness evaluation, we take reference from Figure 15, Figure 16, Figure 17, and Figure 18, and try to synthesize it to understand use continuance of the system. Majority of respondents feel that by listing the examples based on rating does improve persuasive effectiveness of the rating system. Whereas majorities do not agree that ability to rate individual example improves persuasive effectiveness of entire application. Majority of the evaluator also agree that rating system is a nice feature to have in the application. For the final persuasive effectiveness question, we could say majority of the respondent strongly feel that rating system is important to the app's existence.

From the results we received, we could deduce that users feel that the rating system should be present in the support tool. They consider this feature as important feature of the app. The way rating system displays the examples based on rating also is perceived

as effective by the users. Thus we could argue that having rating does have positive impact on user's intention to continue using the rating system. However majority of the users disagree to the idea that, possibility to rate examples improve the persuasive effectiveness of app and thus does not guarantee the intention to use the app as a whole and also the usage of the app.

The respondents who participated in evaluation have prior knowledge of persuasive systems. The results with absolutely no prior experience may have varied opinion regarding the support tool. Also, the number of respondents may have some effect on the result. With the larger audience we can expect some variations in the results. The limited number of examples and fairly lesser amount of activity might have some effect on respondents while responding perceived persuasive effectiveness evaluation.

In general, we received both positive and negative results which assists and guides us to determine future path. The positive responses will encourage us to continue and negative responses will advise us to improve. With the response received it could be argued that the evaluation process has been a success. The essence of evaluating a DSR artifact is to analyze it and improve the areas of weakness. We will use the responses to make the support tool better. Some of the improvement areas have already been identified by usability evaluation, desirability test and perceived persuasive effectiveness evaluation. The semi-structured usability evaluation have inspired us to improve the usability aspects. Feasible improvements in UI and functionality were made during implementation. Whereas, results from desirability test and perceived persuasive effectiveness evaluation will be vital to make improvements in future releases. As a future plan, we will have to analyze the existing feature and add some new ones so that we could persuade users to use it more often. Concept of rewarding the most active users as moderators is already on discussion. Similarly, promoting the profile page of contributor and allowing it to share through social media could be another feature. Depending on the public opinion, adding comments to each example to discuss each example could also be considered. We see this study as a bright and encouraging morning for a beautiful day in persuasive system design area.

6. Discussion

This section attempts to discuss the relevance of this study on persuasive system design field. As Fogg (2003) suggests, with the increase in usage of computer in persuasion, it is imperative to realize the power and use it responsibly. We will discuss various considerations applied during the study. Fogg (2003) envisioned the concept of using computers to change attitude and behavior in ethical and persuasive manner. He also coined the term ‘Captology’ as a field that studies such changes using computers (Fogg, 2003).

The model by Harjumaa and Oinas-Kukkonen (2009) can be considered as a milestone in the field of persuasive system design. They recognized endeavor of Fogg (2003) and used some of his principles to create the PSD model. This model includes design principles divided in relevant and meaningful categories. The model (Oinas-Kukkonen and Harjumaa, 2009) is a manifestation on how to ethically persuade and attain behavioral change. It illustrates the key issues to realize before designing persuasive systems. It also describes the phases and principles to change attitude and/or behavior. PSD model encourages designers and system developers to change the user’s attitude or behavior through persuasion (Oinas-Kukkonen, 2010a).

There have been efforts to use PSD model to create architectures and methods which in turn help the designers to build effective behavior change support systems. Alahäivälä et al. (2013) have proposed an architecture which uses analysis of persuasion context to create BCSS. Oduor et al., (2013) have put forward software design patterns for social influence that uses social support category of PSD model. Drozd et al. (2012) have used design principles of PSD model and hypothesized the relationships to understand perceived persuasiveness of BCSS. Similarly, Lehto & Oinas-Kukkonen (2015) used PSD model to explain and predict perceived effectiveness and use continuance of a BCSS. We could assume that there will be more efforts to use PSD model and its design principle to design and develop persuasive systems. Thus it seems rational and necessary to have a support tool which provides examples of PSD design principles.

In spite of these efforts, the design principles of PSD model have not been utilized to its potential. We identified that there was lack of a system which could provide categorized examples of each design principles of PSD model. We felt that, it could be a reason for PSD model to be underused. Thus we decided to translate this vacuum into our research problem and provide a solution to this barrier to use PSD model with ease. The exponential rise in behavior change systems which are core to the persuasive systems (Oinas-Kukkonen, 2010) also provided us a foundation to pursue the solution to our research problem.

Acknowledging the main research question, our study initiated with understanding the present context of PSD area. Understanding the scenario encouraged us to design and develop a novel artifact to solve our research problem. The artifact was expected to be a specialized tool where persuasive system designers, researchers or anyone in that case can share and obtain precise examples related to persuasive system design principles.

During this thesis, we have studied existing literature related to persuasive systems, architecture and design. The emergence of web 2.0 and availability of smartphones have

dramatically changed the reach and demands of the users. With enhanced hardware and increasing number of devices, we can expect more usage of persuasive systems and more specifically behavior change support systems.

Seven guidelines from Hevner et al. (2004) guided us from design to communication of the study. We have attempted to remain consistent with the guidelines so that the final product would be a novel artifact that serves its purpose (Hevner et al., 2004) This study utilized DSRM to design and develop the artifact to solve a relevant and existing problem of not having a support tool for PSD model. The results from evaluation also support its novelty. This study is consistent to and follows the methodology proposed by Peffers et al. (2007) and attempts to convey the knowledge acquired properly to the targeted audience. We acknowledge that being a support tool for PSD model, it includes a great deal of literature from BCSS. It is because, at present context, BCSS are at the center of persuasive system design. Although not being an outright BCSS, it has utilizes principles from social support category such as competition and recognition. Thus, we attempted to analyze the persuasion context of the support tool using Intent, Event, and Strategy from PSD model.

We analyzed the various software features and functionalities which are of importance to the support tool. The rating system stands out as a feature because of its interactive nature. This also affects how the results are displayed in each principle. In addition, different aspects of support tool were analyzed in accordance to the design principles from the PSD model. This analysis enabled us to understand and deduce requirements about the features and functionalities.

We used 3-tier architecture while designing the support tool. The adoption of layered architecture improves scalability and maintainability of the tool. MVC design pattern was utilized on both front and back ends for consistency and separation of model data from logic and views. Using MVC with layered architecture helps better maintainability of both ends. Angular-fullstack-generator used to create a MEAN stack boilerplate to proceed. Having JavaScript for both frontend and back-end functionality benefitted the implementation in terms of speed as well as limiting the need to learn different languages. Bootstrap was used for UI. Various NPM modules were used libraries as they assist in functionalities.

Evaluation was carried out with the support from group of researchers to test desirability and persuasive effectiveness. We attempted to understand the view of experts in persuasive field. We received 5 responses out of 8 requests sent. The respondents had wide range of expertise level from 1 year to 10 years. The age group of respondents also varied from early 20 to early 40. The aim was to identify desirability and perceived persuasive effectiveness. Identifying desirability would enable us the intangible aspects such as fun and enjoyment. We believe using Product Reaction Cards would be preferred choice to describe subjective facets. It allowed users to describe their feelings about support tool freely rather limiting them to objective answers. Persuasive effectiveness on the other hand allowed us to understand whether users perceive rating system and support tool as effective. We then assessed the responses with the support from models (Drozd et al., 2012; Lehto & Oinas-Kukkonen, 2015) to explain and predict the use continuance and finally the usage of system.

7. Conclusion

This study was an attempt to create a novel system which supports a widely accepted framework in persuasive field. From prior works and existing literature, we identified the need for the support tool which provides examples of design principles categorized in PSD model. We utilized design science research methodology to design, develop and evaluate the tool as it suits such requirements where problem domain demands design, development, and evaluation of novel artifacts.

The major contribution of this study is the artifact and its evaluation. The artifact i.e. the support tool is deduced after careful examination of existing literature. The problem identification and motivation demanded a system to share and obtain examples of PSD principles.

It is a widely accepted concept that persuasive systems should be ethical and hence should not involve coercion or manipulation to attain attitude or behavior change. However, designers may face the dilemma whether the feature they are designing and implementing are ethical. To address this issue, we thought of bringing the community together to share and obtain categorized and ethical examples of PSD model. The support tool utilizes the rating system to involve designers, experts from persuasive field, and novice users to rate the better persuasive examples so that they could take advantage of existing knowledge. This support tool also uses competition from PSD design principle in the sense that it displays the examples based on rating. Listing of examples on the basis of rating will also encourage the users to come up with creative ones for others to implement.

Another important contribution from this could be a linkage between the academic and IS industry. There has been a decent amount of research such as BCSS from academic sector on persuasive systems domain. The IT industry has also been rapid in implementing systems to change behavior of users. For instance, there are abundant applications that measure biometrics such as blood pressure, heart rate, calorie burns, calorie intake etc. We can presume this process will further accelerate in coming days. This support tool is beneficial for industry as the designer and developers can share and obtain examples to implement persuasive principles. The academic sector can also benefit from the reaction of users regarding the principles and its adoption. It is also useful for novice users to compare examples with existing system to analyze the principles used and whether they follow or violate the design principles.

Experts in the field of persuasive systems were involved as respondents for the evaluation of our support tool. We could safely assume that, the knowledge level of these respondents on persuasive systems would be high compared to novice designers. We have to mention that, working on same field in the same research group makes them a homogenous group. If the evaluation process involved respondents from wider group and expertise level, we could have had different results on both desirability and perceived persuasive effectiveness evaluation.

Despite these weaknesses, we could proudly announce that we have a working support tool for PSD model which assists in designing persuasive systems. Moving forward, the tool will be upgraded according to the suggestion from the evaluators and external

contributors. After the upgrade, we plan to release for beta testing using wider audience so that we could have more heterogeneous evaluation. This may enable us to avoid shortcomings that we had during this evaluation. Also the time interval of evaluation could be increased for thorough assessment.

We plan to make this tool as inclusive as possible. We already have some future plans such as promoting the most active and responsible contributors who would be able to moderate the tool.

References

- Alahäivälä, T., Oinas-Kukkonen, H., & Jokelainen, T. (2013). Software architecture design for health BCSS: Case onnikka. *Persuasive technology* (pp. 3-14) Springer.
- Amazon Web Services. (2016). Sign in to the AWS Console [Amazon Web Services page]. Retrieved January 18, 2016, from <https://aws.amazon.com/console/>.
- AngularJS. (2016). Retrieved January 4, 2016, from <https://angularjs.org/>.
- Barnum, C. M., & Palmer, L. A. (2010). More than a feeling: Understanding the desirability factor in user experience. *CHI'10 Extended Abstracts on Human Factors in Computing Systems*, 4703-4716.
- Benedek, J., & Miner, T. (2002). Measuring desirability: New methods for evaluating desirability in a usability lab setting. *Proceedings of Usability Professionals Association, 2003*, 8-12.
- Berdichevsky, D., & Neuenschwander, E. (1999). Toward an ethics of persuasive technology. *Communications of the ACM*, 42(5), 51-58.
- Bootstrap. (n.d.). Retrieved November 8, 2015, from <http://getbootstrap.com/>.
- Casal, J. M. A., Tubío, O. D., Vázquez, R. G., Fernández, M. L., & Yáñez, S. R. (1998). Formalising the software evaluation process. *Computer Science, 1998. SCCC'98. XVIII International Conference of the Chilean Society of*, 15-24.
- Compose. (2016). MongoDB [Compose page]. Retrieved January 6, 2016, from <https://www.compose.io/>.
- Drozd, F., Lehto, T., & Oinas-Kukkonen, H. (2012). Exploring perceived persuasiveness of a behavior change support system: A structural model. *Persuasive technology. design for health and safety* (pp. 157-168) Springer.
- Express (2016). Retrieved November April 24, 2015 from <http://expressjs.com/>.
- Fogg, B. J. (2009). Creating persuasive technologies: An eight-step design process. *Persuasive*, 44.
- Fogg, B. J. (1999). Persuasive technologie301398. *Communications of the ACM*, 42(5), 26-29.
- Fogg, B. J. (2003). Persuasive technology: Using computers to change what we think and do. *Persuasive technology: Using computers to change what we think and do* (pp. 1-282)
- Garlan, D. (2014). Software architecture: A travelogue. *Proceedings of the on Future of Software Engineering*, 29-39.

- Garlan, D., & Shaw, M. (1993). An introduction to software architecture. *Advances in software engineering and knowledge engineering*, 1(3.4).
- GitHub. (2015). Retrieved January October 16, 2015, from <https://github.com/>.
- Green, B., & Seshadri, S. (2013). *Angularjs*. California, USA: O'Reilly Media, Inc.
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly: Management Information Systems*, 28(1), 75-105.
- Intille, S. S. (2004). A new research challenge: Persuasive technology to motivate healthy aging. *Information Technology in Biomedicine, IEEE Transactions on*, 8(3), 235-237.
- jQueryFoundation. (2015). Download jQuery [jQuery page]. Retrieved September 1, 2015, from <https://jquery.com/>
- Jokela, T., Iivari, N., Matero, J., & Karukka, M. (2003). The standard of user-centered design and the standard definition of usability: Analyzing ISO 13407 against ISO 9241-11. *Proceedings of the Latin American Conference on Human-Computer Interaction*, 53-60.
- Karlsson, J., & Ryan, K. (1997). A cost-value approach for prioritizing requirements. *Software, IEEE*, 14(5), 67-74.
- Karppinen, P., & Oinas-Kukkonen, H. (2013). Three approaches to ethical considerations in the design of behavior change support systems. *Persuasive technology* (pp. 87-98) Springer.
- Kennedy, G. A. (1994). *A new history of classical rhetoric*. New Jersey, USA: Princeton University Press.
- Kraft, P., Drozd, F., & Olsen, E. (2009). ePsychology: Designing theory-based health promotion interventions. *Communications of the Association for Information Systems*, 24(1), 24.
- Langrial, S., Lehto, T., Oinas-Kukkonen, H., Harjumaa, M., & Karppinen, P. (2012). Native mobile applications for personal well-being: A persuasive systems design evaluation. *Pacis*, 93.
- LearnBoost, Inc. (2011). Read the docs [mongoose page]. Retrieved September 11, 2015, from <http://mongoosejs.com/>
- Lehto, T., & Oinas-Kukkonen, H. (2015). Explaining and predicting perceived effectiveness and use continuance intention of a behaviour change support system for weight loss. *Behaviour & Information Technology*, 34(2), 176-189.
- Less (n.d.). Retrieved April 10, 2015, from <http://lesscss.org/>.
- Lockton, D., Harrison, D., & Stanton, N. A. (2010). The design with intent method: A design tool for influencing user behaviour. *Applied Ergonomics*, 41(3), 382-392.
- March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, 15(4), 251-266.

- Mardan, Azat. (© 2014). *Pro express.js*. [Books24x7 version] Available from <http://common.books24x7.com/toc.aspx?bookid=78458>.
- Microsoft. (n.d.). Using a Three-Tier architecture Model. Retrieved September 11, 2015, from [https://msdn.microsoft.com/en-us/library/windows/desktop/ms685068\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ms685068(v=vs.85).aspx).
- MongoDB. (2016). Retrieved January 11, 2016, from <https://www.mongodb.org/>.
- Nielsen, J. (1994). *Usability engineering*. California, USA: AP Professional.
- Node.js Foundation. (2015). Retrieved July 9, 2015, from <https://nodejs.org/en/>.
- Nuseibeh, B., & Easterbrook, S. (2000). Requirements engineering: A roadmap. *Proceedings of the Conference on the Future of Software Engineering*, 35-46.
- Oduor, M., Alahäivälä, T., & Oinas-Kukkonen, H. (2014). Persuasive software design patterns for social influence. *Personal and Ubiquitous Computing*, 18(7), 1689-1704.
- Offutt, J. (2002). Quality attributes of web software applications. *IEEE Software*, 19(2), 25.
- Oinas-Kukkonen, H. (2010a). Behavior change support systems: A research model and agenda. *Persuasive technology* (pp. 4-14) Springer.
- Oinas-Kukkonen, H. (2010b). Requirements for measuring the success of persuasive technology applications. *Proceedings of the 7th International Conference on Methods and Techniques in Behavioral Research*, 11.
- Oinas-Kukkonen, H. (2013). A foundation for the study of behavior change support systems. *Personal and Ubiquitous Computing*, 17(6), 1223-1235.
- Oinas-Kukkonen, H., & Harjumaa, M. (2008). Towards deeper understanding of persuasion in software and information systems. *Advances in Computer-Human Interaction, 2008 First International Conference on*, 200-205.
- Oinas-Kukkonen, H., & Harjumaa, M. (2009). Persuasive systems design: Key issues, process model, and system features. *Communications of the Association for Information Systems*, 24(1), 28.
- Peppers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45-77.
- Perloff, R. M. (2010). *The dynamics of persuasion: Communication and attitudes in the twenty-first century* (4th ed.). New York, USA: Routledge.
- Perry, D. E., & Wolf, A. L. (1992). Foundations for the study of software architecture. *ACM SIGSOFT Software Engineering Notes*, 17(4), 40-52.

- Pribik, I., & Felfernig, A. (2012). Towards persuasive technology for software development environments: An empirical study. *Persuasive technology. design for health and safety* (pp. 227-238) Springer.
- Purpura, S., Schwanda, V., Williams, K., Stubler, W., & Sengers, P. (2011). Fit4life: The design of a persuasive technology promoting healthy behavior and ideal weight. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 423-432.
- PuTTY. (2015). Download PuTTY [PuTTY page]. Retrieved June 9, 2015, from <http://www.putty.org/>.
- Quora. (n.d.). Retrieved November 19, 2015, from <https://www.quora.com/>.
- Räisänen, T., Lehto, T., & Oinas-Kukkonen, H. (2010). Practical findings from applying the PSD model for evaluating software design specifications. *Persuasive technology* (pp. 185-192) Springer.
- Selfa, D. M., Carrillo, M., & Boone, M. d. R. (2006). A database and web application based on MVC architecture. *Electronics, Communications and Computers, 2006. CONIELECOMP 2006. 16th International Conference on*, 48-48.
- Simon, H. A. (1988). The science of design: Creating the artificial. *Design Issues*, 67-82.
- Smids, J. (2012). The voluntariness of persuasive technology. *Persuasive technology. design for health and safety* (pp. 123-132) Springer.
- Stack Overflow. (2015). Retrieved November 19, 2015, from <http://stackoverflow.com/>.
- Stormpath. (2015). DOCS [Stormpath page]. Retrieved November July 20, 2015, from <https://stormpath.com/>.
- Tikka, P., Woldemicael, B., & Oinas-Kukkonen, H. (2016). Building an app for behavior change: Case RightOnTime.
- Tilley, S., & Huang, S. (2003). A qualitative assessment of the efficacy of UML diagrams as a form of graphical documentation in aiding program understanding. *Proceedings of the 21st Annual International Conference on Documentation*, 184-191.
- Urgaonkar, B., Pacifici, G., Shenoy, P., Spreitzer, M., & Tantawi, A. (2005). An analytical model for multi-tier internet services and its applications. *ACM SIGMETRICS Performance Evaluation Review*, 33(1) 291-302.
- Venable, J., Pries-Heje, J., & Baskerville, R. (2012). A comprehensive framework for evaluation in design science research. *Design science research in information systems. advances in theory and practice* (pp. 423-438) Springer.
- Venkatesh, V., Thong, J. Y., & Xu, X. (2012). Consumer acceptance and use of information technology: Extending the unified theory of acceptance and use of technology. *MIS Quarterly*, 36(1), 157-178.

Verbeek, P. (2006). Persuasive technology and moral responsibility toward an ethical framework for persuasive technologies. *Persuasive*, 6, 1-15.

W3C. (2014). HTML5. Retrieved May 22, 2015, from <https://www.w3.org/TR/html5/>.

Wiafe, I., Alhammad, M. M., Nakata, K., & Gulliver, S. R. (2012). Analyzing the persuasion context of the persuasive systems design model with the 3D-RAB model. *Persuasive technology. design for health and safety* (pp. 193-202) Springer.

Williams, L. G., & Smith, C. U. (1998). Performance evaluation of software architectures. *Proceedings of the 1st International Workshop on Software and Performance*, 164-177.

YEOMAN. (n.d.) Retrieved March 14, 2015 from <http://yeoman.io/>.

Appendix A. Questions from Evaluation.

First of all, a big THANK YOU for your participation in this evaluation. This evaluation consists of three sections. The first is Background information, second is Desirability test and the third is Persuasive effectiveness evaluation.

All responses will be handled with utmost care and confidence and will not be shared with any third party.

Section 1: Background information

In this section we would love to know a little bit about you. It consists of general questions about you and your professional expertise.

Q1. Your Name, please.

_____.

Q2. Your Age. ($18 \leq \text{age}$)

_____.

Q3. Your Gender.

Male

Female

Q4. Do you use any persuasive technologies yourself?

_____.

Q5. Please let us know your areas of expertise. Select one or more of the expertise areas from the list below. You can also add an expertise area in "other" if your expertise area is not listed.

For example: a) Usability expert for 5 years. b) Persuasive expert for 3 years

Usability Expert

User Experience (UX) Expert

Persuasive Systems/Technology Expert

Human Computer Interaction (HCI) Expert

Software Designer

Software Designer

Add Other

Q6. How many years have you been working in your expertise areas?

Section 2: Desirability test

This section consist of 1 question to understand your feeling after using the application. Please visit the link above and follow the steps from 1 - 6. After the completion of the steps please let us know your experience.

1 Browse through different categories and examples of design principles.

2 From the system, find examples of Recognition from Social Support category.

3 When you find the example of Recognition, click on the example and see if you can see the detail view. From the rating bar in the detail view, give the example some rating based on whether you feel the example you clicked is relevant to social recognition.

4 Navigate to Sign-in / Upload page and click register tab in the top of the navigation bar. Fill the form to register, it will send you a verification link to the email address that you provided. Verify that the registration is complete by clicking on the link in your email.

5 To upload an example, navigate to Sign-in / Upload page and select upload tab from navigation bar. Fill the title and description, select the Recognition from select bar and upload at least two relevant picture of social recognition.

6 See if someone else has rated your example.

Q7. The table below lists a selection of adjectives that can describe a system. Please select 5 adjectives which you think best describe the support tool. Also please let us know the reason behind your selection.

For instance: 1) accessible: I found the information accessible throughout the app. 2) Dull: I thought the user experience was dull.

List of adjectives

The complete set of 118 Product Reaction Cards				
Accessible	Creative	Fast	Meaningful	Slow
Advanced	Customizable	Flexible	Motivating	Sophisticated
Annoying	Cutting edge	Fragile	Not Secure	Stable
Appealing	Dated	Fresh	Not Valuable	Sterile
Approachable	Desirable	Friendly	Novel	Stimulating
Attractive	Difficult	Frustrating	Old	Straight Forward
Boring	Disconnected	Fun	Optimistic	Stressful
Business-like	Disruptive	Gets in the way	Ordinary	Time-consuming
Busy	Distracting	Hard to Use	Organized	Time-Saving
Calm	Dull	Helpful	Overbearing	Too Technical
Clean	Easy to use	High quality	Overwhelming	Trustworthy
Clear	Effective	Impersonal	Patronizing	Unapproachable
Collaborative	Efficient	Impressive	Personal	Unattractive
Comfortable	Effortless	Incomprehensible	Poor quality	Uncontrollable
Compatible	Empowering	Inconsistent	Powerful	Unconventional
Compelling	Energetic	Ineffective	Predictable	Understandable
Complex	Engaging	Innovative	Professional	Undesirable
Comprehensive	Entertaining	Inspiring	Relevant	Unpredictable
Confident	Enthusiastic	Integrated	Reliable	Unrefined
Confusing	Essential	Intimidating	Responsive	Usable
Connected	Exceptional	Intuitive	Rigid	Useful
Consistent	Exciting	Inviting	Satisfying	Valuable
Controllable	Expected	Irrelevant	Secure	
Convenient	Familiar	Low Maintenance	Simplistic	

Section 3: Persuasive effectiveness evaluation

Please follow steps 1-6 from section 2, if you have not already. And let us know about the persuasive effectiveness.

Q8. Listing the items in the order of their rating improves the persuasive effectiveness of the rating system.

	1	2	3	4	5
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

Q9. Possibility of rating articles improves the persuasive effectiveness of the app.

	1	2	3	4	5
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

Q10. Having a rating system makes the app nice to use.

	1	2	3	4	5
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

Q11. Having a rating system is important to the app.

	1	2	3	4	5
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree