DEPARTMENT OF COMPUTER SCIENCE

A Digital Platform for Conducting Research into Social Pledges

Luke Mitchell

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Engineering in the Faculty of Engineering.

Thursday 5th May, 2016

Declaration

This dissertation is submitted to the University of Bristol in accordance with the requirements of the degree of Master of Engineering in the Faculty of Engineering. It has not been submitted for any other degree or diploma of any examining body. Except where specifically acknowledged, it is all the work of the Author.

Luke Mitchell, Thursday 5th May, 2016

Table of Contents

Title Page **1**

Declaration **2**

Table of Contents **3**

Executive Summary **4**
- **Abstract** **4**
- **Summary of Achievements** **5**

Supporting Technologies **6**

Notation and Acronyms **7**

Introduction **9**

Technical Background **10**
- **Gamification** **10**
- **Normification** **11**
- **Open Research Questions in Social Pledging** **12**

Project Execution **12**
- **Project Overview** **12**
- **Software** **13**
  - Design 14
  - Implementation 29
  - Testing 50
  - Documentation 54
  - Summary of Implementation 54
- **Study** **54**
  - Design 54
  - Experimental Method 56
  - Results 58
  - Analysis 66
  - Follow-up Studies 70

Conclusion **71**

Bibliography **73**

Appendix A: Initial UI Testing Checklist **77**

Appendix B: Questionnaire **78**

Executive Summary

Abstract

There is a common intuition that, in sharing a goal with others, you will be more motivated to succeed at it; by taking the step to tell others about an intention the commitment is compounded and the individual is more likely to follow through with their action (Matthews, 2013). A good example of this is with New Year's resolutions: it is easy to imagine that, by telling your friends about your desire to change - say to give up smoking - that you will be more compelled to do so; now your own desire is supplemented by the expectations of others and the wish to succeed. This voicing of an intention to others is known as a "social pledge" and forms the theme that will explored in this dissertation.

There are a number of academic papers exploring the effect of social pledges, and social norms in a wider sense, on participant motivation; there are also a number of theories describing the behaviour underpinning it. The literature is discussed in-depth in the Technical Background section. There is, however, a dearth of rigorous, empirical studies into the adoption and manipulation of social norms to increase motivation, particularly within the remit of digital technologies. This is an oversight that should be addressed and, to do so, a tool was created to allow studies to be performed. These studies will generate invaluable data on the motivational effects of social norms, social pledges and gamification techniques.

An online platform was developed that allows participants to socially schedule and record practice sessions. The platform also contains functionality that analyses any data gathered and allows conclusions about participant motivation and behaviour to be drawn. The platform was implemented in a manner that emphasizes generality and, as such, it can be easily modified to conduct a variety of studies into arbitrary activities.

In an attempt to further understand the interplay between social pledges and motivation, two hypotheses were defined for study: "given a regular task to perform, those that socially pledge their intention to perform it, at a specified time and date, will do so for longer, and more reliably, than those that do not socially pledge." and "given a regular task to perform, those that can see the social pledges of other participants will attempt to match their activity, leading to reduced variance with respect to a group that cannot see the social pledges of other participants."

In this dissertation a study was conducted that attempted to prove these hypotheses, using an instantiation of the generic study platform for studying meditation. Participants were assigned to one of two cohorts: one in which participants were able to interact with one another, pledging socially to meditate; and one in which participants acted in isolation, without the knowledge of the activity of others. The results of the study suggest that the hypotheses are true and that, by pledging socially, motivation to complete a task is improved and a social norm established by participants. Further insights were gained by performing interviews with participants; a key finding is that scheduling a task dramatically increases the motivation to complete it. Several recommendations for the design of digital communities wishing to increase participation have been made as a result of these findings.

By creating the platform in a generic manner the possibility of conducting a raft of follow-up studies is both possible and, with respect to the requisite infrastructure, trivial. The platform was created with ease of customisation in mind, and with the aim that a novice user can quickly and easily set up their own studies. To demonstrate this, details of several follow-up studies have been provided and the steps required to perform customisation of the platform have been given.

Summary of Achievements

The outcomes and achievements of this dissertation can be summarised as follows:

1. The creation of a generic, enterprise-standard platform for performing studies into social norms, social pledges and the motivation of task performance.
2. The completion of a specific study, using the aforementioned study platform, to research a particular example of digital technology for motivation, which makes a scientific contribution to the field of Human-Computer Interaction.
3. The creation of an enterprise-standard platform for the analysis of data gathered using the aforementioned study platform that supports the use of parametric and nonparametric testing and generates meaningful graphs for analysis.
4. The creation of documentation supporting the aforementioned platforms, serving as reference material for study participants and enabling third-party researchers to use it for future studies.
5. The creation of three open-source *NPM* packages containing basic statistical methods and for performing parametric and nonparametric testing.
6. The creation of two open-source *NPM* packages that provide the only available JavaScript implementations of Levene’s test and the Brown-Forsythe test.
7. The design of several follow-up studies, confirming the findings of the study and investigating the similarities between social pledges and social completions, have been described, utilising the aforementioned platforms with minimal modification.

Supporting Technologies

The following supporting technologies were used:

- I used the NPM module `async` to provide asynchronous helper methods.
- I used the NPM module `bcryptjs` to provide a hashing function for user authentication.
- I used the NPM module `body-parser` to parse form data.
- I used the NPM module `cookie-parser` to parse cookie data.
- I used the NPM module `debug` to provide debugging information.
- I used the NPM module `express` to provide a framework for defining routes and serving content.
- I used the NPM modules `grunt`, `grunt-bower-concat`, `grunt-bower-task`, `grunt-cli`, `grunt-contrib-jshint`, `grunt-contrib-uglify`, `load-grunt-tasks` to automate the build and deployment process.
- I used the NPM module `jade` to performing templating when serving content.
- I used the NPM module `jsonwebtoken` to manage API tokens and authentication.
- I used the NPM module `moment` to manipulate date objects.
- I used the NPM module `mongo-sanitize` to sanitize input data when querying the MongoDB database.
- I used the NPM module `mongoose` to query the MongoDB database.
- I used the NPM module `mongoose-deleted` to handle soft deletion within the MongoDB database.
- I used the NPM module `moniker` to generate random user credentials for the study platform.
- I used the NPM module `serve-favicon` to cache and serve the favicon.
- I used the NPM module `validator` to check for valid URIs.
- I used `bower` to automate dependency management.
- I used the `fullcalendar` library to display schedules.
- I used the `jQuery` library for front-end JavaScript.
- I used the `bootstrap` library as a CSS framework.
- I used the `jquery.cookie` library for manipulating cookies from the front-end.
- I used the `jquery-ui` and `jt.timepicker` libraries for displaying timepicker components from the front-end.
- I used the `html5shiv` and `jquery-placeholder` polyfills for displaying modern components on legacy browsers.
- I used the `font-awesome` library for displaying icons.
- I used the `Montserrat` and `Lato` web fonts.
- I used the `Heroku` platform-as-a-service provider and the `mLab` database host to freely host the study.

Notation and Acronyms

The following acronyms are used within this document:

- AJAX: Asynchronous JavaScript and XML
- ANOVA: Analysis of Variance
- API: Application Program Interface
- CSS: Cascading Style Sheets
- DNS: Domain Name Service
- DOM: Document Object Model
- EU: European Union
- HCI: Human-Computer Interaction
- HREF: HyperText Reference
- HTML: HyperText Markup Language
- HTTP: HyperText Transport Protocol
- HTTPS: HyperText Transport Protocol (Secure)
- IANA: Internet Assigned Numbers Authority
- ID: Identifier
- JS: JavaScript
- JSON: JavaScript Object Notation
- MEAN: MongoDB, Express, Angular and Node.js
- NoSQL: Non-SQL
- NPM: Node Package Manager
- REST: Representational State Transfer
- SSL: Secure Sockets Layer
- SQL: Structured Query Language
- TLS: Transport Layer Security
- URI: Uniform Resource Identifier
- URL: Uniform Resource Locator
- XHTML: Extensible HyperText Markup Language
- XML: Extensible Markup Language

Acknowledgements

I would like to thank the many willing participants I recruited for the study for their help with my research and for their invaluable feedback when developing the study platform. I would also like to thank my supervisors Dr Chris Preist and Dr Daniel Schien for their ongoing help and support.

Introduction

Interest in gamification, the idea of using game design principles in non-game contexts, and the use of social norms to motivate behaviour are relatively new phenomena (Google, 2016). The growth of these areas, however, has been explosive, with a huge number of start-ups and organisations employing the techniques: from *Weight Watchers* (Weight Watchers UK, 2016) to *Duolingo* (Duolingo: Learn Spanish, French and other languages for free, 2016), applications from a diverse range of fields are incorporating social pledges, points-systems and competitive elements in their products. Gamification is a technique that has been in use since the late 1990's and has been of interest to researchers since the early 2000's; the benefits of which are often touted. The manipulation of social norms however, particularly within digital communities, is a technique that has yet to be fully explored by researchers; despite the fact that it becomes ever more relevant as we move towards a world driven by social media.

The use of game elements in applications, a technique termed gamification, is widespread and has gained purchase in several key areas: educational tools, crowdsourced scientific data-mining, fitness trackers and, more recently, corporate performance trackers. The last of these comes with the emergence of *Bunch Ball* (Bunch Ball), a product that integrates with the corporate operations product *SAP* (SAP Software Solutions, 2016), allowing businesses to gamify their targets by providing badges, points and leaderboards to rank employees.

Through gamification, or participant interaction in a wider sense, social norms can be established within a digital community[1]. Social norms dictate the behaviour of community members and research suggests that people feel a strong urge to conform to these norms, particularly when they feel that their behaviour will be observed and emotionally judged by other community members. A technique called normification can be used to inspire the adoption of new social norms, either within an existing situation or when creating novel experiences. This can be used to encourage people to act by increasing their motivation to behave according to these social norms.

A social pledge is the action of voicing a commitment, or an intent, in a social manner; this can be in person or digitally, via social media or otherwise. There is evidence to suggest that pledging an intention socially can have a dramatic effect upon the motivation of the pledger to fulfil their pledge[2] and, throughout this dissertation, this is the theme that will be explored.

[1] Digital communities range from employees within a business to users of an electronic message board or members of a social network; almost everyone using the Internet is a member of one such community.
[2] An example is the website Promise or Pay (Promise or Pay, 2016) in which users socially pledge an intention and, if they break their intention, they must pay their pledged amount to a specified charity. The website reports that their method combines two approaches that increase the likelihood of fulfilling a pledge by 33% and 72% respectively.

Technical Background

Gamification

Gamification, a term first coined in 2002 (Marczewski, 2012), is defined as “the use of game-design elements in non-gaming contexts” (Deterding, 2011). The use of gamification, as briefly discussed in the Introduction, has been rapidly gaining prominence as a motivational technique (Massung, 2014).

An example of gamification from the literature is GamiCAD (Li, 2012), a gamified tutorial for the *AutoCAD* software taking inspiration from two commercial equivalents: *Microsoft Ribbon Hero* and *Adobe LevelUp*. In their paper, Li et al describe a tutorial that has been developed using gamification techniques to guide a new user through several examples. The user earns points for each task they complete and they are rewarded with encouraging sounds and visual stimuli. The study found that all users of the tutorial completed a test task faster than control users and, when providing qualitative feedback, indicated that the experience was more enjoyable, more engaging and easier than reported by control users.

Gamification tends to be competitive, making use of leaderboards, points and badges that can be compared to those of other participants within an application. These techniques are discussed, in-depth, by Glover (Glover, 2013). The use of gamification often leads to an increase in motivation for high-achieving participants but can lead to a decrease in motivation by average or low achievers. Qualitative data on this effect has led to a belief that such users feel as though they were being “left behind” (Preist, 2014) and the at the problem was insurmountable.

Preist et al explore the use of competitive gamification in the Close the Doors case study (Preist, 2014) and found that some participants excelled through competition, typically those that were high-achievers, but that low-achievers found the leaderboards and comparison discouraging. High-achievers often described the leaderboard as motivating further participation when other participants were close to displacing their rank or had been particularly active. The study also found unexpected results such as participants that used the leader board to ensure that they were performing on average or above and that, beyond this, their performance was not of personal significance.

Collaborative forms of gamification also exist and have been explored, particularly within an educational context. This has resulted in an increase in performance in some studies (Tally, 2012) (Ferenstein) (Simões, 2013) but it has been argued that, due to the inherent competition in many gamification techniques, its use reduces motivation to complete the task itself and only increases the motivation to succeed in the newly established competition (Duffy, 2010).

A paper by deMarcos et al (de-Marcos, 2014) describes a study in which undergraduate students completed online tasks, alone and in collaboration with others, to gain trophies and badges. The study explored the difference in performance of groups completing the tasks individually and collaboratively, using a control group, without access to the tasks, as a comparison. The study found that both groups completing the online tasks performed better than the control group and that the collaborative group performed slightly better than the individual group. The study also found that the control group performed better for the final exams, however, which was unexplained.

Normification

Normification is defined as the adoption of new social norms, either within an existing situation or when novel experiences are created, and is seen as a non-competitive form of gamification. The concept is described by three theoretical models: the Focus Theory of Normative Conduct (Cialdini, 1990), the Theory of Normative Social Behaviour (Rimal, 2005) and the Norm Activation Model (Harland, 1999). The Focus Theory of Normative Conduct states that a normative belief can be *descriptive* or *injunctive*, relating to whether it describes the behaviour of a social group or the approval of behaviour by a group. The Theory of Normative Social Behaviour states that individuals aspire to variously attach or distance themselves from social groups and will modify their behaviour accordingly, aiming to match their perceived behaviour of a desired group. The Norm Activation Model states that, in addition to these perceived group norms, individuals have personal norms that significantly impact their behaviour. These personal norms are made up of the individual's feelings and moral obligations towards a particular behaviour and may be encouraging or discouraging.

By manipulating a perceived social norm the behaviour of individuals can be modified**.** This can be leveraged to encourage a desired change, such as shops closing their doors during the winter months (Preist, 2014), users finding and reporting software bugs (GetBadges - gamification platform for developers, 2016), or participants performing a difficult or repetitive task such as revising for exams.

Schultz et al (Schultz W. K., 2008) found that providing descriptive information about the behavioural norm led to an increase in performance by under-performing participants but a decrease in performance by over-performing participants, with both approaching the mean; this finding was mirrored by Chen et al (Chen, 2010). The effect, in relation to high-performing users returning to a point of average behaviour, is known as the 'boomerang' effect. Chen et al also found that the increase was greater than the decrease, however, leading to an overall improvement in performance.

When both descriptive and injunctive beliefs are manipulated the effect on behaviour is increased, with respect to manipulating the perceived descriptive beliefs alone (Schultz W. K., 2008)**.** An example, provided by Schultz and Nolan (Schultz P. N., 2007), is to include an emoji to indicate a positive or negative report when describing the energy usage of a household, modifying the injunctive norm of participants. The emojis used were a smiling and a frowning face, indicating happiness and unhappiness with the report: the smiling emoji was used when the household behaviour was better than the norm whilst the frowning emoji was used when behaviour was worse. Schultz et al found that including an emotion in the report was sufficient to counteract the 'boomerang' effect, preventing participants performing above the norm from reducing their efforts and becoming complacent.

The desire to be acknowledged for positive behaviour plays an important role in motivating behaviour, as suggested by Rotman et al (Rotman). When behaving in a manner that agrees with a perceived injunctive norm participants wish to make their behaviour known to the group. Rotman et al suggests that the motivation for volunteers participating in scientific studies changes over time: the initial motivation stems from scientific curiosity but transitions to a desire for acknowledgement of their contributions.

Finally, Gollwitzer (Gollwitzer, 1986) found that, when completing a set task, completing a substitute task dramatically reduces the motivation to complete the original task. It is suggested that the substitute ‘completion event’ replaces the desire to complete the original task. The completion of a task is seen as required and so becomes the descriptive norm; the specific task to be completed, however, is seen as superfluous to this. Mahler suggests that this is more apparent when the substitute completion event is social, such as when it is observed or verified by another person (Mahler, 1933).

Open Research Questions in Social Pledging

The attempted manipulation of social norms to induce behavioural change has been explored in several studies, as mentioned in the preceding section. The behaviour surrounding social pledges - the act of members in a group voicing their intentions - and the effect this has yet to be explored and poses several interesting questions: will the group adopt a new norm based upon these pledges, as suggested by Postmes et al (Postmes, 2000), and are group members more likely to fulfil their intentions after pledging them socially?

An additional question that warrants exploration is whether pledging an intention socially has a similar effect to a social completion event: does pledging an intention to fulfil a task negate the need to actually fulfil it? This question builds on work by Gollwitzer and Mahler and could provide crucial insight into the construction of platforms that leverage social pledges to maximise performance.

Finally, performing a study into the effects of social pledging on performance of participants would provide an insight into the effectiveness of the technique as a social motivator. By exploring the performance of participants repeatedly performing a task, and pledging their intention to do so, the effect on the overall performance, of the group and individual participants, could be discovered. Additionally, the presence or absence of the boomerang effect, as described by Schultz and Nolan, could be observed.

Project Execution

Project Overview

In order to explore the relationship between perceived social norms and an individual’s motivation to perform a task a study was designed that would enable participants to make social pledges: voicing their intention to perform an activity or action, at a specified date and time. The study was designed to draw comparisons between the effectiveness of those pledges when they were made socially, to a cohort of other participants, versus pledges made privately, by an individual using the platform solitarily.

Software

To facilitate the study a generic, enterprise-standard platform for performing studies into social norms and the motivation of task performance was designed and implemented. The platform was created in such a way that it is accessible to as many users as possible - simply requiring an Internet connection and relatively new browser[3]. The platform also utilises current web technologies ensuring it is performant and easily maintainable by individuals with experience working in the field.

The software allows its users to interact with one another whilst pledging to perform an arbitrary, specifiable task. The users can pledge individually or opt to join other users and can log their activities in real-time or retrospectively using a simple tool. The platform also offers a condensed, easy-to-digest overview of the activity of its users, known as the "feed"; this enables users to quickly gauge the amount of activity undertaken by other users, an important feature in developing a notion of a social norm.

An administrative component was also created, allowing all aspects of the platform to be modified in real-time by a non-technical user. The activity on the platform can also perform statistical analysis using a secondary component known as the *analysis platform* (with respect to the *study platform*). The analysis platform is integrated into the administrator control panel and provides statistical methods for performing parametric and nonparametric analysis on the amount of activity recorded by various user groups.

Throughout this section, and the remainder of this document, the study software as is referred to as "the platform". Elsewhere, the platform software has been named "Zen" and this may be found when looking for it online or within documentation. Further, the instance of the platform used to conduct the study was also named "Zen" and this title can be seen in the screenshots of the software. The final design, with this naming convention, can be seen in Figure 1. To emphasize the generality and customisable nature of the platform the generic terms of *action* and *activity* have been used; this is opposed to *meditation*, the specific action chosen for the study.

A set of credentials for the study platform have been created to allow readers to explore the software. The first set of credentials is a mock study participant whilst the second is an administrator. These are given below.

Credentials: https://boiling-ravine-11892.herokuapp.com/
`alice : finalproject7878`
`overseer : conquertheworld1212`

Credentials: A participant (top) and an administrator (bottom) for the study platform.

[3] The platform is hosted as a website and has been tested on browsers as old as Internet Explorer 9, released in 2011.

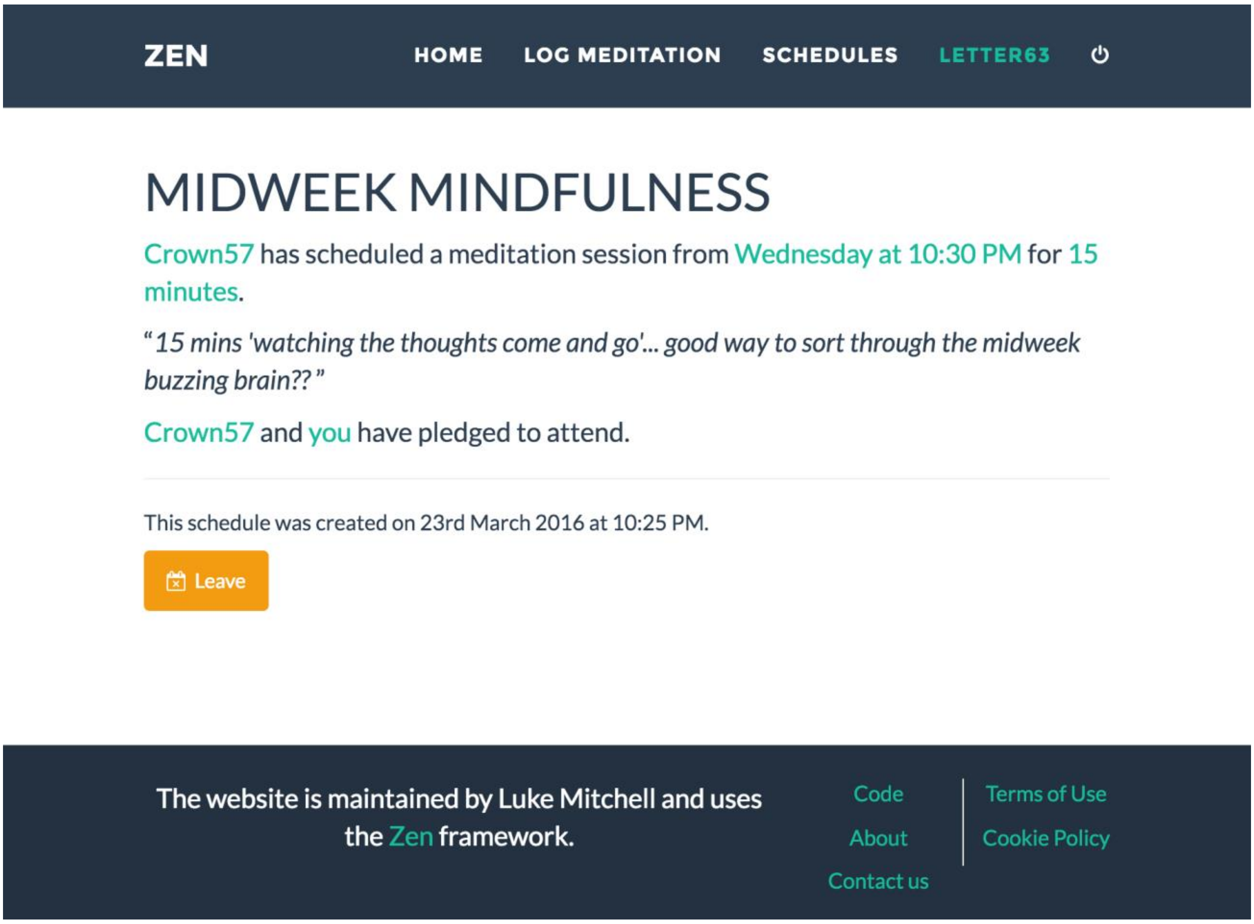


Figure 1: The final platform design, as seen on a wide-screen browser. This page shows a scheduled meditation, created by the participant *crown57* and pledged to by the viewing participant, *letter63*.

Design

When designing the study platform several key tenets were identified: ease of use, simplicity and adaptability. These tenets were chosen to maximise the possible impacts of the platform, both in terms of the meditation study and for possible follow-up studies, to be performed by the author or by third-parties, once this dissertation has been concluded.

Keeping the platform easy to use was a priority as an ideal study should include a wide and varied participant demographic. It is often the case that certain demographics, such as senior citizens, have a low level of computer literacy and, without careful design, they could easily be alienated from the study due to difficulty in using the platform. To combat this careful consideration was made regarding the user experience at each stage of the iterative design process; during this time user feedback was gathered from a variety of would-be participants.

The simplicity of the platform was also paramount to its success: by keeping the design of the platform simple and modular it would be much easier for a third party researcher to use it for future studies, a secondary goal to it’s creation. The design was tailored such that elements could be modified, added or removed with ease, and that a developer familiar with the technologies used, but new to the platform, could familiarise themselves quickly and without undue effort.

Additionally, the concept of simplicity extends beyond the underlying code. By keeping the user interface and the functionality of the platform simple the user's learning curve and the number of independent variables could be reduced. To achieve this a single feature was opted for where several could be included[4]. This had the benefit of easing the task of writing documentation, reducing the time spent testing and reducing the volume of code.

Finally, the value of the platform is that it allows a large variety of studies to be performed; keeping the software adaptable and generic ensures that this is possible. A platform was envisioned where a simple configuration, performed by an administrator or researcher, could generate a platform tailored to an array of unique studies - preferably with little or no requisite modification to the software.

It was decided to create the study platform as a web application, exposing all participant functionality via a website and all programmatic functionality via a back-end API. The decision to use a web application as opposed to a mobile or desktop application was primarily made due to the availability of websites across multiple platforms: users of all major operating systems, both desktop and mobile, can access websites with minimal aesthetic variation; implementing the platform in another format would have required the creation of multiple versions and would have created the additional complexity of ensuring that their function was identical. Cross-platform solutions such as Java exist, however, due to the familiarity most computer users have with accessing web applications, it was felt that this presented a simpler solution.

Functionality

The platform is conceptually simple: an application that allows its users (hereafter "participants") to pledge their intention to perform an activity, to log their completion of the activity and to view the pledges of other participants; additionally, it would allow participants to join the pledges of others', creating a many-to-one relationship. To formalise this the following definitions were created:

1. Schedule: *a date and time that an action is to be performed, as created by a participant*
2. Pledge: *a commitment made by a participant to perform an action at a time specified by a schedule*
3. Fulfilment: *a record or log of an action performed by a participant, as part of a pledge or independently*
4. Action: *an activity or task performed by a participant as part of a study*

[4] An example of this is the real-time logging feature; during initial user feedback a "countdown" option was requested, allowing the user to set a desired length of time to perform their action, after which an alarm sounded. This would have added an additional variable to the study being performing and so the author declined to implement it.

This formalisation allows the requisite functionality for the platform to be clearly identified. The core function of the platform is to allow participants to schedule and complete activities, with additional functions enabling them to view and join the schedules of other participants. The activity of other participants must also be prominent, allowing their pledges and fulfilments to be seen by other platform users, allowing social norms to develop within the community.

When scheduling a practice, a function central to the use of the platform, it was decided that taking inspiration from popular scheduling tools such as *Microsoft Outlook*, *Google Calendar* and *Apple Calendar*; these tools are used on a daily basis by many people, often integrated with an email client or present on a smartphone, and their interface is known and seen as standard. The interfaces to these tools are similar and, once one has been used, understanding the function of any other is trivial. By replicating such an interface participants using the platform would instinctively understand how to make schedules and would ensure that the platform is easy to use. A comparison of the interfaces of two of these tools can be seen in Figure 2.

The conventional interface for a scheduling tool is to employ a digital calendar that can be scrolled through and viewed in several modes: monthly, weekly and daily. The scheduled events are marked as boxes on the calendar, occupying the days and times their specified days and times, represented as a position within a particular day. To view a schedule an event is clicked upon, bringing up more detailed information about it, such as a start and a finish time, a description, the ability to set reminders and so forth. Events may be edited similarly, by clicking on them to show the detailed view, then clicking an *edit* button. To schedule a new event, the start date and time are selected by clicking the corresponding area of the calendar followed by an *add* button.

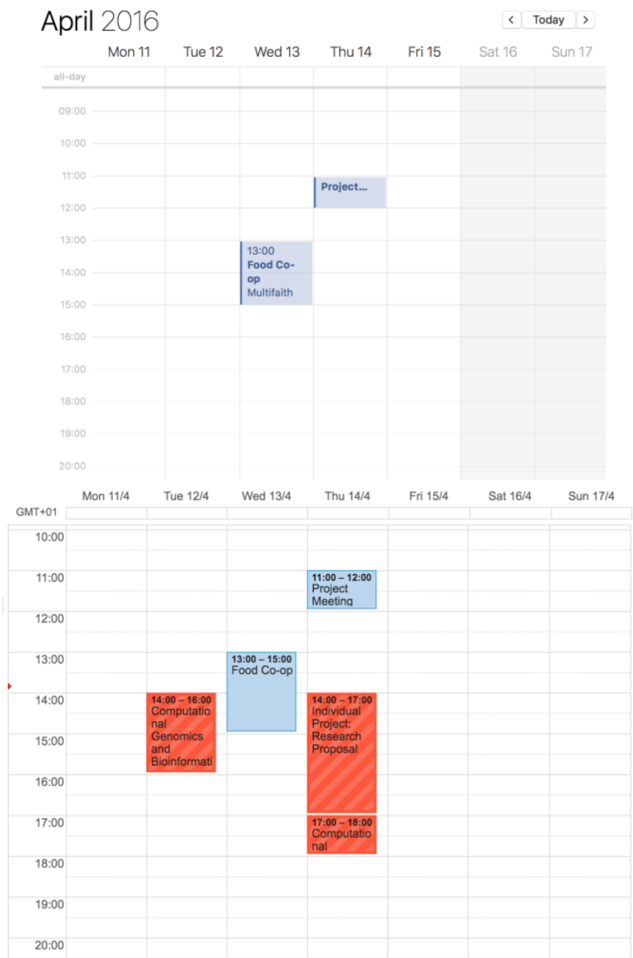


Figure 2: A comparison of the *Apple Calendar* and the *Google Calendar* interfaces.

In keeping with the conventional interface, the platform displays participant schedules on a calendar, shown as coloured boxes occupying the space between their start and finish times. All other functionality has been kept true to this design: multiple views; the ability to view and edit schedules by clicking on them; and by adding schedules by clicking on the desired start day or time, or by clicking the dedicated *add* button. As participants require the ability to pledge to attend the schedules made by other participants additional buttons to *pledge* and *leave* are included in the detailed schedule view, alongside a list of pledged participants. When viewing the schedules in the calendar the pledge status of the participant is shown by the colour of the schedule box: green schedules have been pledged to whilst red have not; this behaviour can be seen in Figure 3.

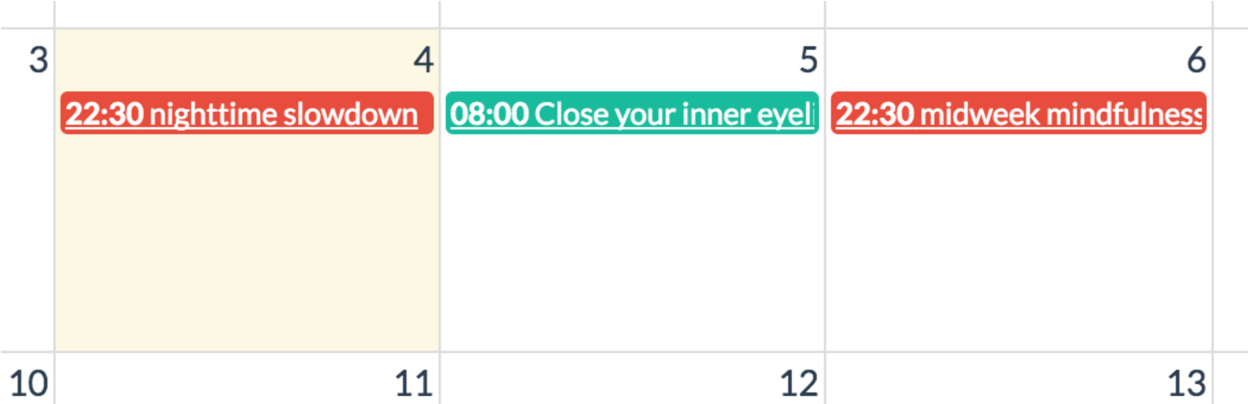


Figure 3: Scheduled events shown in the platform calendar, as seen by a participant. Schedules shown in green have been pledged to whereas schedules shown in red have not.

Another crucial function is the ability for participants to log their actions and to fulfil their pledges to schedules. This is achieved through a *Log* page that provides the participant with some basic statistics about their usage of the platform and gives them the option of *Log 'action'* or *Perform 'action' now*[5]. The participant can then click the appropriate button depending upon whether their intention is to record an action retrospectively or in real-time using the platform.

To record an activity retrospectively the participant simply enters a start date and time, filled in using two simple *picker* components[6], and an end date and time. The end date is automatically set to the start date once one has been selected, although this can be manually adjusted if the activity spans multiple dates. The end time is also automatically set, defaulting to 15 minutes after the start time.

If a participant wishes to fulfil a pledge to a schedule in retrospect then they can use the *Log 'action'* button found in the detailed schedule view. The *Log 'action'* button is present for past schedules that the participant has pledged to but not yet fulfilled; pressing the button auto-populates the date and time fields with those of the schedule. This feature was requested by a participant during an initial feedback session as they commented that they were "not able to be at a computer when meditating" and that "it seems like a lot of hassle to enter the same information twice".

To record an activity as it is performed, in real-time, the participant is able to use the platform's timer. This function allows the participant to press a *Start* button when they are ready to begin and to have a timer track the time they spend performing their activity. Upon completion a *Finish* button can be pressed, confirming the entry, or a *Cancel* button can be used to delete it.

---

[5] As mentioned in the introduction to the Software section, these buttons actually read *Log Meditation* and *Meditate Now*. This text is generated from a customisable dictionary that can be used to quickly change the focus of a study and, to emphasize this generality, the terms *Log 'action'* and *Perform 'action now'* have been used in the discussion.

[6] A *picker* is a common name for a component displaying a selection of values, from which the user can select or *pick* one. Examples of pickers include time and date pickers, as used within the platform, or cities and countries.

If the participant is performing a scheduled activity then this information will be presented to them whilst on the 'real-time' page: the name of the schedule, it's start time and it's end time will be visible. Additionally, the other participants that have pledged to the schedule will be shown alongside an indication of whether they are online. This updates dynamically keeping the participant informed of how long the schedule has left and the status of the other pledged participants. Additionally, a gentle *chime* sounds when a schedule begins or ends, alerting the participant to the completion of their schedule even when distracted[7].

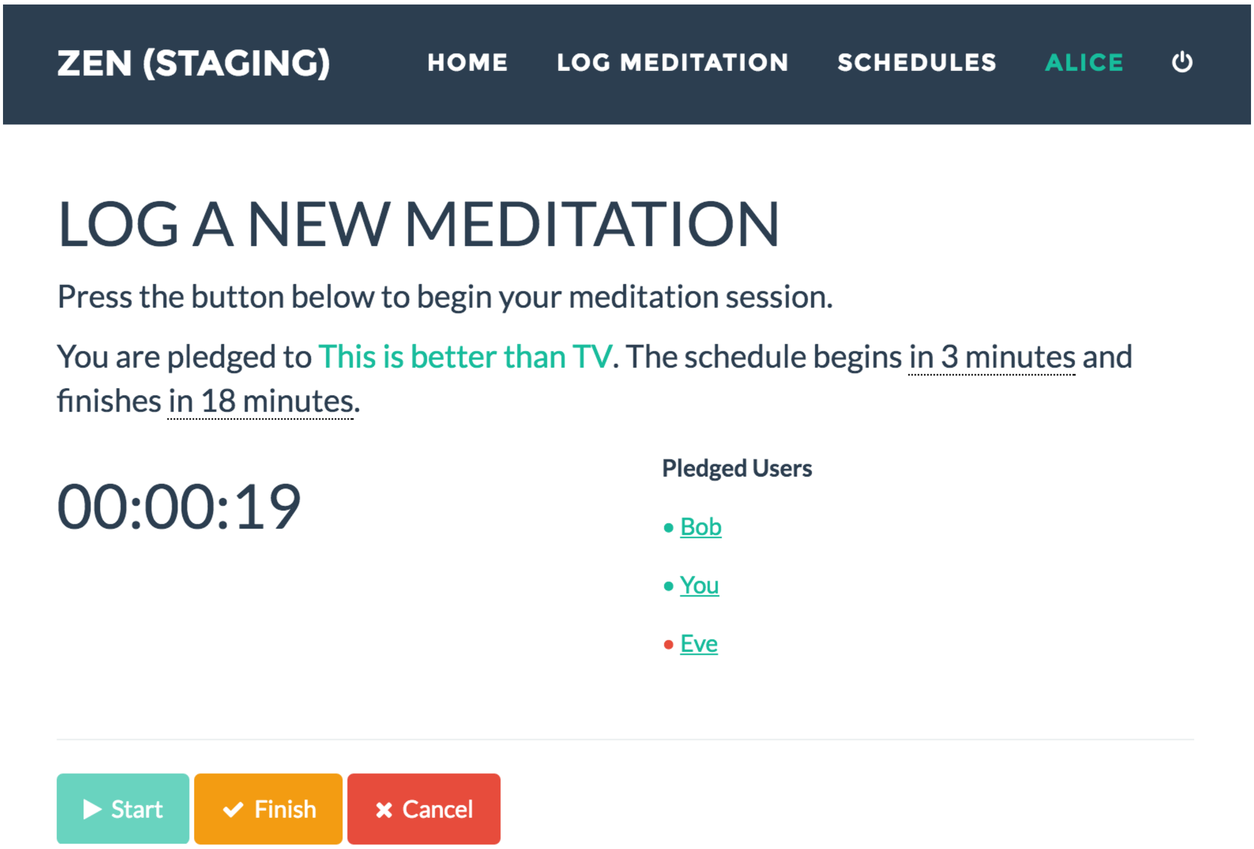


Figure 4: Logging an action in real-time, as seen by a participant. The participant has pledged to attend the schedule *This is better than TV* and information about it is visible; the exact start and end times can be seen by hovering over or tapping the underlined approximations. Another pledged participant, *bob*, is online whilst a third pledged participant, *alice*, is offline.

The activity of participants is amalgamated as a feed, serving as the *home page* for signed in participants. The feed, taking inspiration from the *Facebook "news feed"*, takes the schedules and fulfilments of participants, including the creation of schedules and pledging to those created by others, and presents them individually, ordering them in descending order by the creation date. This provides an up-to-date view of the activity within a group that can be easily understood. By taking a familiar approach to the display of this data, as with the

[7] This is particularly useful for the study that was performed as the activity is meditation. During the practice of meditation it is common to close your eyes and so watching a computer screen will be impossible and a distraction.

interface for the scheduling component, it was hoped that the difficulty faced by new users to the platform would be reduced.

The items displayed within the feed are represented differently, allowing a participant to quickly gain information about the activity and differentiating them from one another. Additionally, each type of item is given an icon, allowing it to be identified at a glance. The information displayed has been *humanised* and will replace the viewing participant's username with "you", choose the correct tense and supporting verb when enumerating lists and choose the correct tense depending upon whether the subject is past, future or present.

The motivation for including a feed stemmed from a desire to allow participants to be aware of the activities of other participants, reinforcing the idea of the social norm set by participant behaviour. Without a condensed, amalgamated view such as one offered by the feed this is a piecemeal task and difficult to achieve - let alone motivate a participant to perform. Using an activity feed as the homepage alleviates this as the participant is exposed to the feed upon signing in and will be presented with an overview of the preceding days in an easy to digest form.

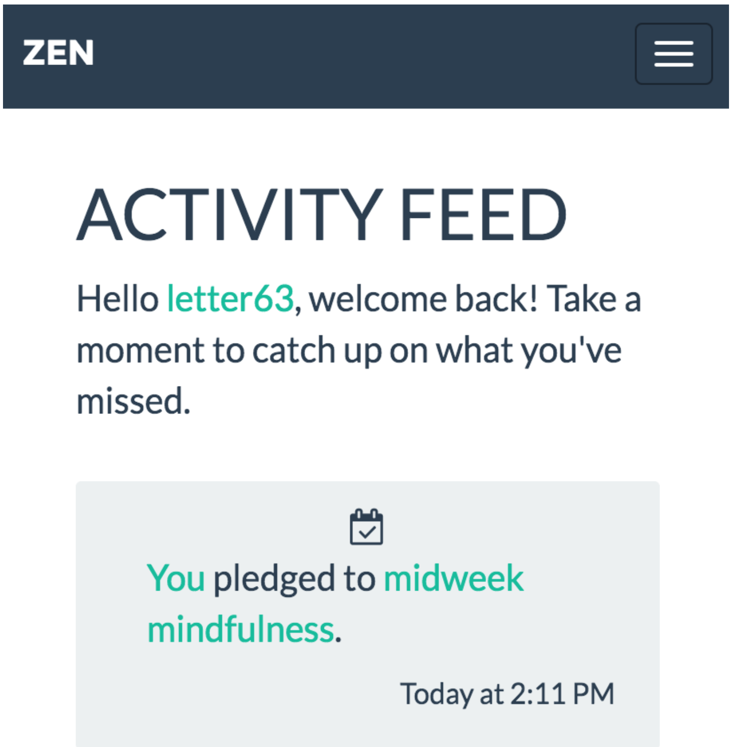


Figure 5: The activity feed, as seen by a participant using a mobile browser. The feed shows an amalgamation of all the activity of participants within their group.

Database Entities

Once the functionality for the platform had been formalised the data entities became obvious: *schedules*, *pledges*, *fulfilments*, *users* and *groups*. A participant, or a *user*, can create *schedules*, *pledges* and *fulfilments*. The *user* is also a member of one or more *groups*; this governs what is visible to them as they use the platform, for example, by restricting the visibility of activities, participant accounts and schedules to participants in the same group.

Entity Name	**Attributes**	**Comments**
User	Username, password, groups, administrator	The administrator boolean provides access to the control panel and associated API endpoints.
Group	Title, description	A description is optional.
Schedule	ID, Title, Description, Start, End, *User*	A description is optional.
Pledge	ID, *Schedule*, *User*	-
Fulfilment	ID, Start, End, *Users*	This entity is used to record activities both as part of a pledge or independently. See below for details.

Table 1: Entities and their attributes. Primary keys are shown underlined whilst foreign keys are shown in *italics*.

One problem emerged when designing the database entities: participants needed to be able to store information about their activities regardless of whether it was associated with a *pledge*. Due to this requirement there needed to be a more general way to log activities rather than simply marking a *pledge* as complete. To combat this the *fulfilment* entity includes start and end timestamps and is not linked to a *pledge* via a foreign key; several queries have been written that find *pledges* made by the user for *schedules* that occur during the *fulfilment* period, making this functionality available.

A many-to-many relationship exists between *users* and their *groups* as it was desirable for the platform to support participants to be members of multiple groups, deeming this important to the flexibility of future studies. Many-to-many relationships traditionally require an intermediary entity, to which both original entities have a one-to-many relationship; this behaviour can be seen in Figure 6.

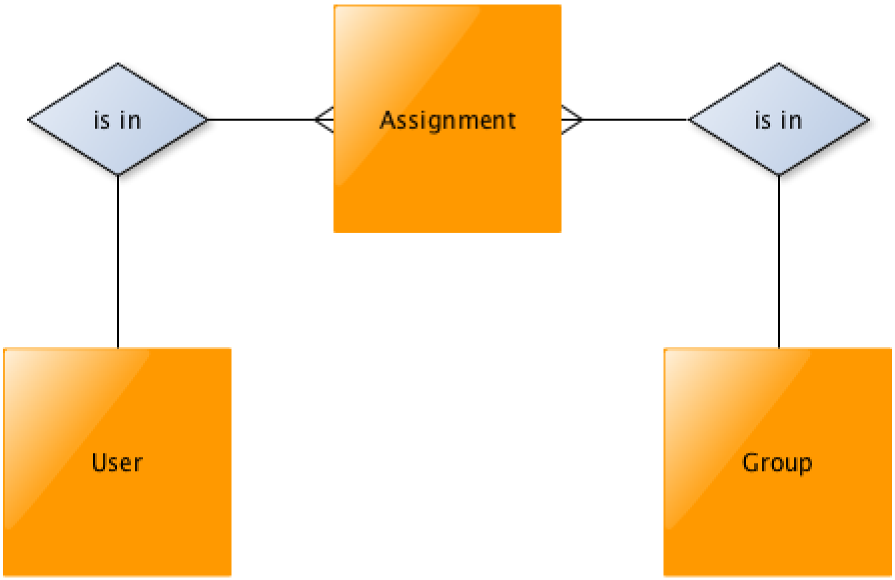


Figure 6: Traditional entity-relationship diagram for a many-to-many relationship.

Due to the use of the MongoDB database in the implementation of the platform, which is discussed in more depth in the Implementation section, the additional *assignment* entity is not necessary. Using a document-oriented database such as MongoDB allows storage of data that is less rigidly defined and which can be added to, or modified, with ease. The *user* schema stores an array of *groups* and the *assignment* entity is omitted.

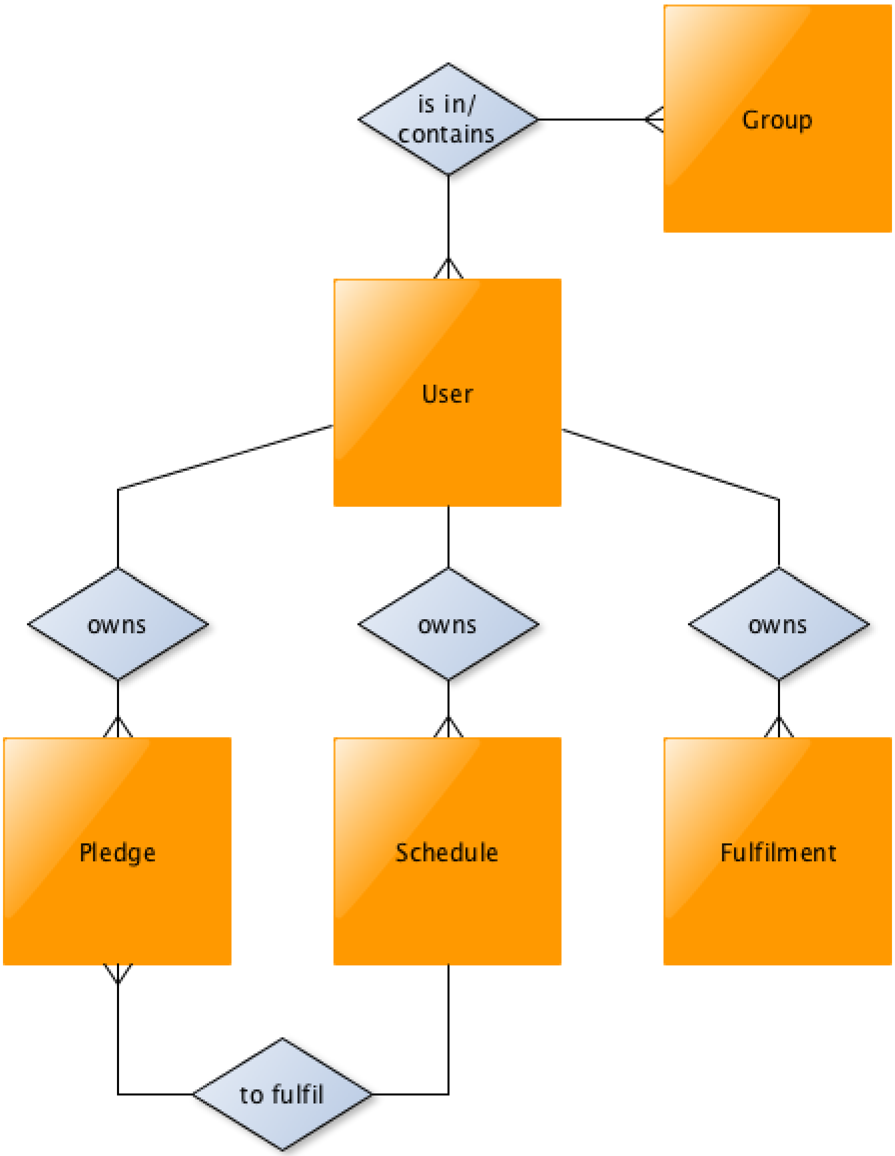


Figure 7: The final entity-relationship diagram for the platform.

Analysis Platform

When performing a study, the researcher needs to be able to analyse it. This is typically done by performing statistical tests on the data gathered and comparing the results of various samples, between which the *independent variable*[8] has been modified. When performing research using the study platform these samples will correspond to groups of participants, with the *dependent variable* being the amount of time spent performing the studied activity. To ease the task of the researcher an analysis platform was designed that allows various statistical tests to be performed on different groups.

[8] In statistics there are two types of variable: independent and dependent. The independent variable is altered by the researcher to produce an observable effect on the dependent variables.

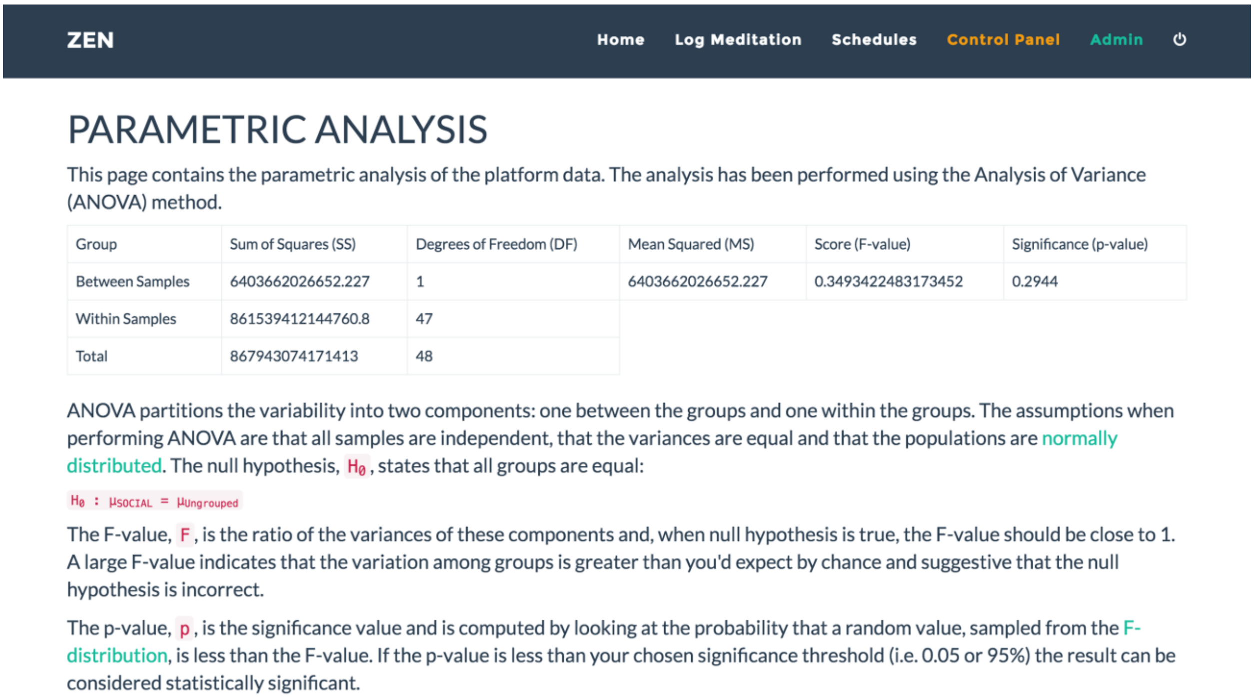
The analysis methods commonly used can be broadly categorised into *parametric* and *nonparametric*, with the former making the assumption that the data is *normally distributed*[9]. In the case that a relatively small sample size is used it cannot often be assumed that the data is normally distributed and so nonparametric methods are employed. Additionally, it can occasionally be helpful to view both types of statistical test to gain insight into the data. The analysis platform has been designed so that both parametric and nonparametric tests may be performed, allowing the user to specify which groups to compare and which comparison method to use. By performing these tests a significance value is computed; this indicates the likelihood that any correlations appearing in the data are the result of chance and are insignificant.

A common parametric method is the analysis of variance, often shorted to ANOVA; this is a robust method for performing hypothesis testing across multiple samples by attempting to subdivide the samples into groups that, collectively, fit the distribution of the original data. The inclusion of ANOVA is important as it has been cited as the "most used statistical technique in psychological research" (Howell, 2002)**.** A common nonparametric method is the Mann-Whitney U Test, a statistical test that allows two groups to be ranked and a value, named the U value, to be produced describing them. The implementation of both methods is discussed in more detail in the Implementation section.

The analysis component of the software was integrated into the administrator control panel for ease of use, available via an easy to find link. An overview of the study platform is given presenting common statistics such as the mean times scheduled, pledged and logged, the number of each and the range of responses; an overview of individual participants or groups can also be viewed by selecting the name from the respective drop-down box. To perform statistical tests comparing multiple groups the analyst can select the name of the group from two adjacent drop-down boxes, similar in appearance to that of many *diff* tools[10]. Once the group names have been selected a statistical test can be chosen and the analysis tool will display the results alongside a brief explanation of their meaning.

[9] Data is normally distributed when it can be approximated by a *Gaussian distribution*. Normally distributed data is clustered around a mean and falls sharply at either side, forming a bell-shaped curve when plotted on a graph.

[10] A diff tool is used to compare different versions of a file and will highlight each modified line, or change in line, of the two selected versions or filenames. The interface for graphical diff tools often incorporates two adjacent drop-down boxes for selecting the versions or filenames to compare.


ZEN Home Log Meditation Schedules Control Panel Admin

PARAMETRIC ANALYSIS

This page contains the parametric analysis of the platform data. The analysis has been performed using the Analysis of Variance (ANOVA) method.

Group	Sum of Squares (SS)	Degrees of Freedom (DF)	Mean Squared (MS)	Score (F-value)	Significance (p-value)
Between Samples	6403662026652.227	1	6403662026652.227	0.3493422483173452	0.2944
Within Samples	861539412144760.8	47			
Total	867943074171413	48			

ANOVA partitions the variability into two components: one between the groups and one within the groups. The assumptions when performing ANOVA are that all samples are independent, that the variances are equal and that the populations are normally distributed. The null hypothesis, H_0, states that all groups are equal:

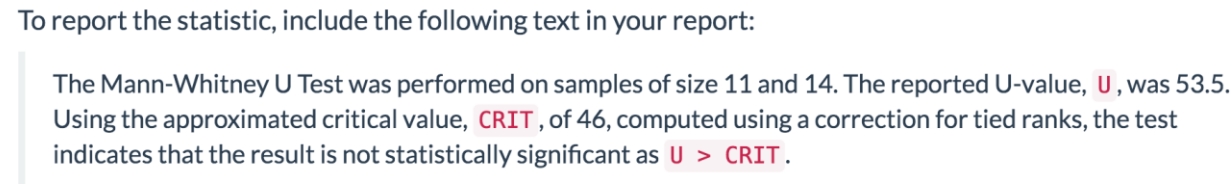
$H_0 : \mu_{SOCIAL} = \mu_{Ungrouped}$

The F-value, F, is the ratio of the variances of these components and, when null hypothesis is true, the F-value should be close to 1. A large F-value indicates that the variation among groups is greater than you'd expect by chance and suggestive that the null hypothesis is incorrect.

The p-value, p, is the significance value and is computed by looking at the probability that a random value, sampled from the F-distribution, is less than the F-value. If the p-value is less than your chosen significance threshold (i.e. 0.05 or 95%) the result can be considered statistically significant.

Figure 8: The parametric analysis page showing the results of the meditation study. The table shows the data required to describe the analysis of variance and the text explains how to interpret the data.

The display for the statistical tests, additionally to providing details for how to interpret the resulting values, gives an estimated interpretation. This is displayed in a format that can be used in a report and contains all the figures that must be stated. An example of this can be seen in Figure 9.


To report the statistic, include the following text in your report:

> The Mann-Whitney U Test was performed on samples of size 11 and 14. The reported U-value, `U`, was 53.5. Using the approximated critical value, `CRIT`, of 46, computed using a correction for tied ranks, the test indicates that the result is not statistically significant as `U > CRIT`.

Figure 9: An interpretation of the nonparametric statistical test, made by the analysis platform.

Customisation

The ability to customise the platform to suit a large array of studies into social norms, social pledges and gamification was of utmost importance - second only to the platform enabling the meditation study to be performed. By creating the platform in a generic manner it would extend its remit of use far beyond this dissertation, ideally into becoming a tool employed by researchers in the area. One of the project’s supervisors, Dr. Preist, had voiced his desire to perform multiple studies of a similar nature and indicated that it would be useful if the platform could accommodate that.

Initially the behaviours that would typically need to be changed for a study were listed: database credentials; administrator credentials; platform name; organisation name and contact details; *about us* text; and the action being studied. Additionally, the behaviours of participants, such as their ability to view the actions of other participants, or their ability to interact with others, would need to be customisable.

To ease the task of customising the platform, from the perspective of a future researcher, it was decided to use a single configuration file[11], from which the behaviour of the software could be defined. The configuration file contains the details required to connect to the database being used for the study and the initial credentials for the administrative user, the details for whom will be added to the database the first time the software is launched[12]. The configuration file also contains seed strings, used to increase the security of hashing functions or random number generation when storing user passwords and issuing API tokens; these must be changed for security purposes and there is a comment in the file indicating as much. These options may also be specified using environment variables which offers an increase in security when storing sensitive information like passwords. A full list of configuration options has been given in Table 2.

A dictionary is also included in the configuration file. The dictionary allows the activity being performed to be quickly and easily changed: just by specifying the noun, in singular and plural, and the verb, in present and past tenses. The dictionary also contains entries for replacing the words for *schedule*, *fulfilment* and *pledge*. The dictionary, along with many of the other configuration options, is passed to the templating engine and used to generate the content for the platform; changing the dictionary will result in every reference to "meditation" becoming whatever new action has been specified.

Configuration Option	**Description**
Name	The name of the study, used in page titles, to create the home page and for the terms of service and cookie policy pages.
Organisation	The name of the organisation maintaining the website. This will usually be the university or organisation performing the study.
Email	The email address of the organisation maintaining the website. This is used to generate the contact page.
Locale	The location of the study, used for formatting time and date strings.
Database	This specifies database connection credentials, such as the MongoDB URI.
Token	The secret string used to seed the API token generator.

[11] The file is `config.json` found in the project's root directory.
[12] The software will only create the administrative user the first time the software is run *unless* a special file, `install.lock`, is removed prior to the running it subsequently. A message to this effect is presented when the software is run.

Admin	The initial credentials of the administrative user, created at install.
Pages	This specifies the directory within which to search for additional content pages. The parameters used to display the pages are also included, although optional.
Links	This specifies additional navigation or footer links to include.
Dictionary	The phrases used to describe the studied *activity*, *fulfilment*, *pledge* and *schedule*.

Table 2: Configurable options included in the configuration file.

The inclusion of additional content pages, such as those describing the study, or providing additional instructions, is achieved by placing a template in a special *pages* directory, defined in the configuration option of the same name. All additional content pages found in the *pages* directory are then parsed and made available to participants using parameters specified in the configuration file, if present, or sensible default values. The parameters allow content pages links to be added or removed from the navigation bar, including an option to display for *guests only*; links to them to be added to the footer and the position specified; the title to be specified; and the URL *slug*[13] that the pages are served from to be specified.

Several default pages are included in the directory that have been designed to serve most purposes; these include an about page, a contact page, a cookie policy and a terms of service page. The content of these pages is generated from the contents of the configuration file, using the organisation name, the platform name, the contact details and the dictionary. If there is a requirement or a desire to modify them, however, they can easily be altered by a future researcher.

Other Considerations

There was a raft of additional considerations when designing the platform, most of which stemmed from a desire to create a tool that was production-ready; as such everything required to launch a study had to be included with the software.

When using cookies[14] on a website available within the European Union (EU) a visible notice must be present on every page, allowing the user to opt-out - or informing them that this is not possible and that, by continuing to use the website, they are agreeing to its cookie policy. As the study being performed for this dissertation was taking place within the EU this was a concern; a *cookie bar* that conforms to the legal requirements was implemented to comply to this legislation.

A generic cookie policy was found and adapted to suit the needs of the platform. The cookie policy, similarly to the notice described above, is subject to EU requirements and must name and describe all cookies used by the website. A generic terms of use page was also found and

[13] A slug is a section of the URI after the domain name, such as an identifier or page name.
[14] Cookies are small files that are saved to the user's browser when visiting a website, often used to store state or preference information.

adapted to suit the platform. Both the cookie policy and the terms of use include the organisation name and contact information, retrieved from the configuration file, and require no modification before publishing a study.

The configuration file has been designed to parse environment variables, overwriting the default values present. This feature was included so that a study can make the software used, the platform as-is or in modified form, open-source without having to worry about removing hard-coded credentials or other configuration information. This feature is particularly useful when deploying the platform to *Heroku* (Cloud Application Platform | Heroku, 2016) or other platform-as-a-service (PaaS) providers[15]. The platform is easily deployable to *Heroku* and instructions have been provided for doing so in the documentation.

Finally, a fully-integrated control panel was included with the application. This provides tools for the creation and modification of users, which may be study participants or other administrators, and for the modification of all schedules, pledges and fulfilments made on the platform. The control panel also allows groups to be created and applied to users, allowing the users within the same groups to interact with one-another. A screenshot of the administration panel can be seen in Figure 10.

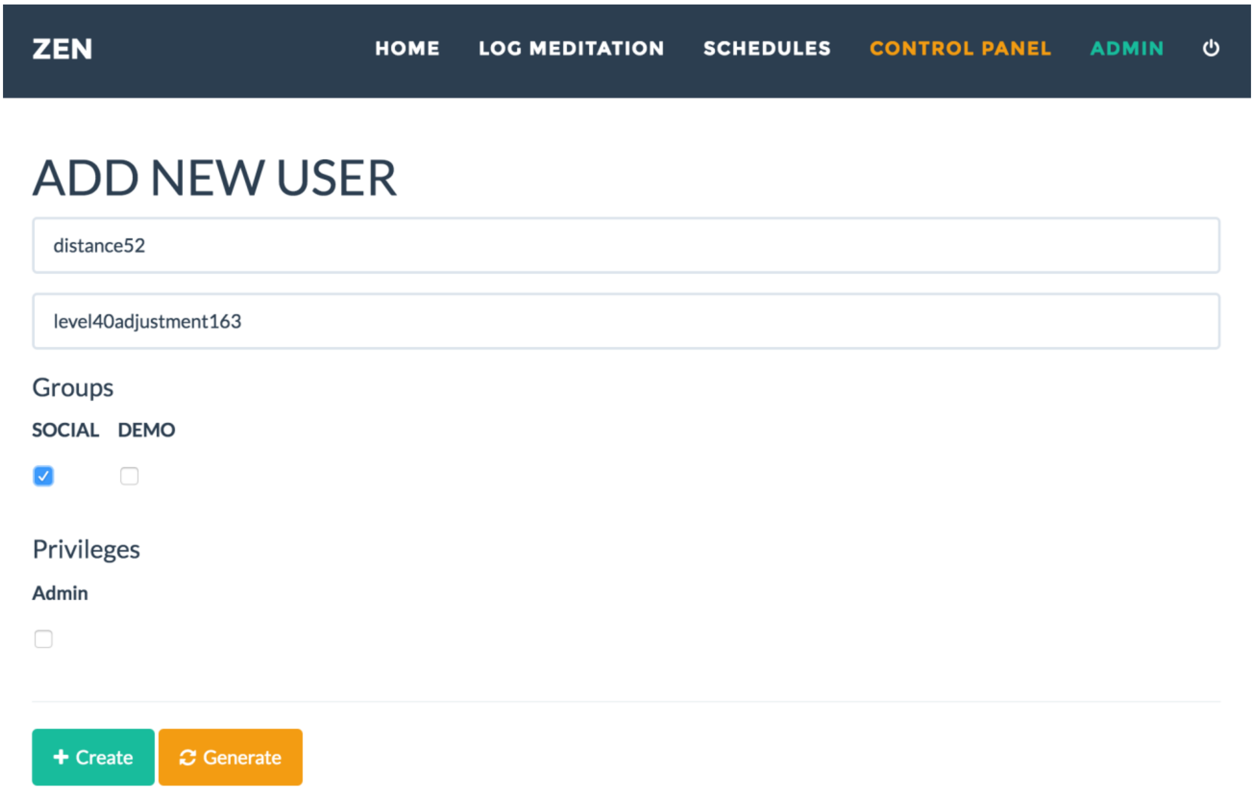


Figure 10: The administrative control panel showing the form used to add new participants. The participant being created is a member of the SOCIAL group.

[15] Platform-as-a-service is a type of cloud offering that enables users to publish their applications or products in a manner that recognised by an online platform. This typically involves using a subset of languages with a platform API for certain aspects of their functionality.

The control panel can be accessed via a link in the navigation bar, visible only to signed-in administrators. The control panel presents the administrator with a list of options and allows them to browse the users, schedules, fulfilments, pledges and groups respectively. The control panel also allows the creation, deletion and modification of each of the options. When creating a user, the username and password may be generated automatically, utilising a friendly word list to create memorable combinations; this allows anonymous alias' to be given to participants with ease.

Design Criteria

Once the initial design phase had been completed the following criteria for the study platform were developed. These formed the basis of my design and, along with the mock-ups and feedback from initial testing and discussions with users, were used as a template when implementing the study platform.

1. The platform must include the ability for participants to create schedules
2. The platform must include the ability for participants to pledge to attend schedules created by other participants
3. The platform must include the ability for participants to fulfil schedules, either in real-time or retrospectively
4. The platform must present activity data to the participant in an easy to interpret "feed"
5. The platform must allow the creation of participant groups, within which activity data is visible to all members
6. The platform must allow participants to be created in isolation of a group and that the activity of these participants is private to them
7. The platform should allow features that can be toggled per-group to modify the study environment
8. The platform must feature a control panel for administration of users and activities
9. The platform must be easy to use and intuitive for participants
10. The platform should be easy to use and intuitive for administrators
11. The platform must be accessible to all participants with a recent browser and operating system
12. The platform must be visually customisable with little difficulty or effort
13. The platform must be generic and configurable to an arbitrary task
14. The platform must be deployable with little difficulty or effort

Additionally, the following design criteria were developed for the analysis platform:

1. The platform should provide an overview of the study
2. The platform should offer parametric statistical testing
3. The platform should offer nonparametric statistical testing
4. The platform should be easy to use and intuitive for administrators
5. The platform must be deployable with little difficulty or effort

Implementation

The software, both the study and analysis platforms, were implemented using *Node.js*. Node.js is a lightweight, asynchronous, single-threaded JavaScript environment with a huge number of libraries and packages available for it (Module Counts, 2016); it has come to public attention in recent years and is the industry choice for many products (Enki, 2016) (Node.js, 2016) (Behr, 2013). Using JavaScript for both the back-end and the front-end is advantageous as it reduces the number of technologies required for using and understanding the platform; this, in turn, reduces the complexity of deploying the platform, as per study criterion 14 and analysis criterion 5, and the level of technical expertise required to maintain it, as per study criterion 13 and analysis criterion 4.

The use of Node.js for web applications has also become commonplace in recent years (Voyer-Perrault, 2016) with several common software-patterns (known as stacks) being employed. It was decided to use a variation on the *MEAN* stack (MEAN.io - The Friendly & Fun Javascript Fullstack for your next web application, 2016), *MongoDB, Express.js*, the *Jade* templating engine and Node.js; this omits Angular.js in place of a static templating engine due to the nature of the platform. A justification for using these technologies is given below.

MongoDB is an industry-standard NoSQL, document-oriented database; NoSQL is not suitable for all applications due to only guaranteeing *eventual consistency*, however, as the data gathered by the study platform is not timing-critical - it is not necessary that all participants see the same information at *exactly* the same time - it can be used for this software. An advantage of using a document-oriented database is that it ensures that modification to the schema does not break existing code, as documents are not required to be homogenous: documents within the same collection can be structured differently. This is important as it allows modification by third-party researchers without requiring extensive re-writes, fulfilling study criterion 13 and analysis criterion 4.

Further, MongoDB is lightweight and easy to configure, even for an inexperienced user, also fulfilling study criterion 13 and analysis criterion 4. The database system has been designed for the cloud and can be deployed in clusters, allowing the software to be scaled indefinitely. Finally, there are providers such as *mLab* (MongoDB Hosting - Database-as-a-Service by mLab, 2016) offering free, secure MongoDB deployments that can be used for conducting studies with minimal setup.

Express.js is a Node.js framework that is often employed for creating *RESTful* applications[16]**.** Using the framework allows API endpoints to be defined easily using neat, intuitive code. The Express.js framework also allows custom authentication handlers to be defined and supports a wide array of rendering techniques.

Jade is a fast, simple to use templating engine that integrates easily with Express.js. A templating engine renders an HTML template, using data specified in JavaScript object notation (JSON), to be served to the user. This process improves the load-time with respect to using asynchronous JavaScript and XML (AJAX) to query an API from simple pages. Another advantage of using Jade is that layouts and page components can be shared and included as they are required, reducing code duplication and increasing modularity.

[16] REST, which stands for representational state transfer, is an architectural paradigm for creating APIs that are used to create, modify and retrieve content to display. Applications using a REST API are termed “RESTful”.

Angular.js was omitted as most content is static and there is little or no need for the dynamic views or updating provided by angular. Where dynamic elements are required, which is a minority of pages, they can easily be implemented using AJAX and manipulating the elements being displayed[17] without the additional bulk and complexity of the Angular library.

Finally, Node.js was chosen, as mentioned above, to minimise the number of technologies utilised within the software and for its lightweight footprint. The use of Node.js and Express.js results in a high-performance, single-threaded web server that can handle a large number of concurrent connections; the performance of the platform has been discussed in more depth in the Performance section, found later in this dissertation. Due to being single-threaded the issue of thread-safety is not an issue, reducing the complexity of developing using the language and alleviating the task of testing the resultant application. Additionally, Node.js is a language supported by *Heroku*, a platform-as-a-service provider offering free instances; by keeping the platform lightweight and using Heroku-compatible tools I provided a simple and easy way to deploy the platform, a valuable trait for both the study performed in this dissertation and for future studies, as per study *criterion 14* and analysis *criterion 5*.

Specification and Support

The front-end of both platforms aims to conform to the newest *HTML5* and *CSS3* specifications. These are current industry best-practise and allow for the widest range of features and the most readable code. Opting to use the most recent specifications also *future-proofs* the platform to a large extent as, whilst it is unlikely that *HTML* or *CSS* will implement any breaking changes[18] in their evolution, programming best-practises may change and by using the most recent recommendations, these are likely to stay current for longer.

The combination of HTML5 and CSS3 leads to more compact, easier to understand code; this is important for code maintainability and for the desired goal of being easily modified by a third party researcher, as per *criterion 13*. It also increases the likelihood of becoming a top result for search engine queries, although this is less of a priority due to the intended use of the platform (Uchicha, 2012).

As per study *criterion 11*, the front-end was designed to fall-back gracefully when a client browser does not support its features - such as those introduced with *HTML5*. The platform has been designed to support browsers as old as *Internet Explorer 9*. The decision to use *Internet Explorer 9* as a baseline was due to it representing a small fraction of traffic[19] without requiring large code rewrites; older versions of Internet Explorer have negligible usage and are now unsupported by Microsoft[20] (Support for older versions of Internet Explorer ends. Microsoft, 2016).

[17] This is known as modifying the domain object model or the DOM.

[18] A breaking change is a modification to a specification that results in products built with previous versions becoming unusable or unstable. This is extremely unlikely to occur with *HTML* and *CSS* as the effects of breaking old websites would be disastrous.

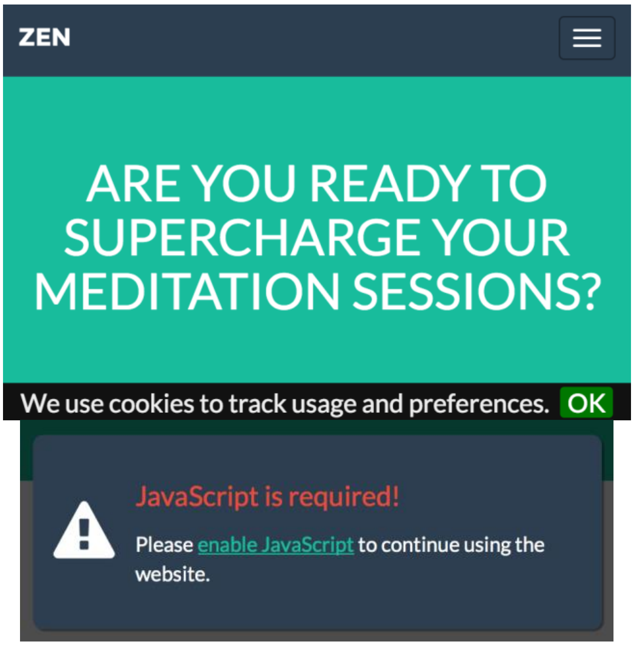
[19] *Internet Explorer 9* is currently used by around 0.3% of users visiting w3schools.com. The actual usage may be slightly higher as non-technical users are more likely to use older browsers and unlikely to visit the website, but it can be taken as a rough figure.

[20] Versions of Internet Explorer older than 11 are no longer supported by Microsoft.

Older browsers are supported through the use of *shivs* and *polyfill*: these are small JavaScript files that are included in the page and emulate recent functionality on older, unsupported browsers. One example of where this is employed is to mimic the `placeholder` HTML attribute, which is not present in *Internet Explorer 9*; the placeholder is a short text string displayed within input boxes and form components before the user has entered a value. During initial user testing the omission was discovered and a fixed was implemented through the use of a *polyfill*.

The front-end of the study platform also makes use of AJAX to create a dynamic experience, for example by refreshing the activity feed or online status of participants, and to perform initial validation of data entry, reducing server load and the improving the user experience. Using JavaScript to perform asynchronous retrieval of data has several benefits including reducing the initial load time of pages, allowing the dynamic update of feeds and real-time logging and the ability to respond quickly to user selections, such as retrieving schedules to populate the calendar.

Due to this, JavaScript is a requirement for users of the platform. This was deemed acceptable as only 1.1% of users typically disable JavaScript (Herlihy, 2013) and, through the addition of a notice to enable the feature via a `noscript` HTML tag, users are given instructions to enable the feature or update their browser. This behaviour can be seen in Figure 12.



Figures 11 and 12: Messages displaying requisite cookie message and a warning message to user's browsing without JavaScript.

Study Platform

The study platform is implemented as a standalone web server that serves a front-end user interface, generated by populating a template with content queried from a database, and exposing an API utilised by the front-end. The web server and API functionality leverages the Express.js framework to define both static pages and dynamic routes, allowing the URI to be parameterised. Routes (an API endpoint and an associated behaviour) have been defined for schedules, pledges and fulfilments, each with pages to display and modify the data accordingly. A route has also been defined for the administration control panel and for the analysis platform. Each route has custom authentication middleware; a method is called each time the route is requested that will perform checks to ensure the requester is authenticated and, in some cases, transform the response into an appropriate format. The middleware will respond to unauthorised users or malformed requests with the appropriate HTTP status code (IANA, 2016) and a human readable message.

The pages served are parsed using the Jade templating engine: the template is specified along with a JavaScript object containing information used to construct the page. The templating engine allows for files to be broken into modules and inserted into one another; this feature has been used to specify header, footer and layout elements that are included from all pages. The templating engine also allows iteration functions, such as `for each`, to be used; this condenses the code required to display unknown amounts of data into several lines.

The database used to store participant data is MongoDB. The database is document-oriented and does not, by default, restrict the data stored in each collection; this functionality can be implemented by defining *models*, allowing types, default values and validation to be specified. A model has been created for schedules, pledges, fulfilments, users and groups, each containing validation and appropriate default values. Static and instance methods were also defined for each model, allowing common procedures, such as retrieving the members of a group, to be called easily in an object-oriented fashion.

Configuration

When implementing the study platform Node.js was chosen, for reasons detailed above. A typical Node.js application is configured via JSON or a JavaScript object in a file in the root directory; in keeping with this, the study platform is configured using a JavaScript object defined in `config.js`. The configuration file contains parameters that allow the study activity, the associated vernacular and the display of all peripheral application content to be altered. The configuration file was included to allow modifications to the study platform to be made without technical knowledge of the software: all that is needed is the documentation provided and a text editor.

Contained within the configuration file is the name of the platform and the name, and contact details, of the organisation hosting it. A dictionary, specified as a JSON object in the form `key:term`, is included that contains the word for the studied action (i.e. ‘meditation’), the schedule, the pledge and the fulfilment. The words are defined for several tenses and used to generate written content throughout the platform: each time an activity-specific word is used it is included from the dictionary object. Various security options are included: database credentials, a secret string for *salting*[21] API tokens and the initial administrative user credentials. Other options, such as the study locale can be specified; this is used to format

[21] A salt is a random string added to a plaintext before encryption or hashing to defend against dictionary attacks and rainbow-table lookups.

time and date strings in the convention used in the selected locale. These options can be seen in Table 2.

The study platform parses the configuration file at start-up and checks the validity of the parameters. The parameters are then stored in memory; any changes require the application to be restarted. The configuration is cached in memory as it is passed to the templating engine and used to render every page served by the application. As such, loading it from the file for each request would be a major performance bottleneck. The study platform also checks for the existence of a file lock, `install.lock`; if the lock is not present the platform attempts to create an administrative user, using the credentials specified in the configuration. This behaviour allows a known username to be created when the platform is first run. If the user has already been created the password is reset to the one specified in the configuration, allowing access to be restored in the event of the administrator losing their password. The file lock is recreated each time this step is performed, ensuring that unauthorised users cannot re-run the script and gain access.

By ensuring that the application is highly-configurable, and that this can be quickly and easily performed, I have fulfilled study design criterion 13. Simple modifications, such as rebranding or recontextualising the activity studied, can be achieved by changing as little as 5 lines of code. In Code Listing 1 an example configuration is provided for a platform studying the motivation behind soccer practice: the platform is named 'Soccer+'[22] and a fictional organisation has been used.

[22] The Soccer+ platform has been deployed and can be found online (Soccer+, 2016).

Code Listing: JavaScript

```
name: 'Soccer+',
organisation: 'Soccer Plus',
email: 'contact@soccerplus.com',
dictionary: {
 action: {
  noun: {
      singular: 'soccer',
      plural: 'soccer'
  },
  verb: {
   present: 'play soccer',
   past: 'played soccer',
      presentParticiple: 'playing soccer'
  }
 },
 fulfilment: {
  noun: {
      singular: 'practice',
      plural: 'practices'
  },
  verb: {
   present: 'practise',
      past: 'practised'
  }
 }
}
```

Code Listing 1: The configuration options needed to generate the 'Soccer+' example.

The appearance of the platform can also be easily modified in the sample configuration file via the `pages` and the `links` parameters. By altering the `nav` Boolean a page can be added to the navigation bar and, by setting the `footer.display` and `footer.position` parameters, the page or link can be added to the footer in a specified location. The style and the static page text can all be changed with a minimum of effort; this is achieved by modifying or replacing `style.css` (as described below) or adjusting the templates found in the `pages` directory.

The styles used to display the platform utilise the *Twitter Bootstrap* (Twitter Bootstrap, 2016) CSS framework and a modified theme (Freelancer Theme, 2016). Where required, additional styles have been defined in a single file, `style.css`, and have been designed to work in harmony with Bootstrap. The advantage of using an existing, mature CSS framework is twofold: it avoids 'reinventing the wheel' for common classes, such as those used to display content within a grid and governing the display over multiple screen resolutions; it also allows a new theme to be quickly and easily applied from the large number of freely-available Bootstrap themes, allowing the platform to be re-skinned with a minimum amount of work. Both of these advantages are important as they increase the availability of the platform for participants, as the experience when using mobile browsers is consistent and well-tested, and they make it fast and simple to customise the appearance of the platform. These advantages satisfy design criteria 11 and 12, respectively.

The theme can typically be altered by changing the `bootstrap.css` file and making minor alterations to `theme.css`, or, in some cases, removing it. A simple change[23] using a third-party theme (Cerulean Theme, 2016) can be seen in Figure 13. The modifications involved downloading an open-source theme in place of `bootstrap.css` and defining a custom footer in theme.css. A total of 36 lines of CSS were written for the change[24].

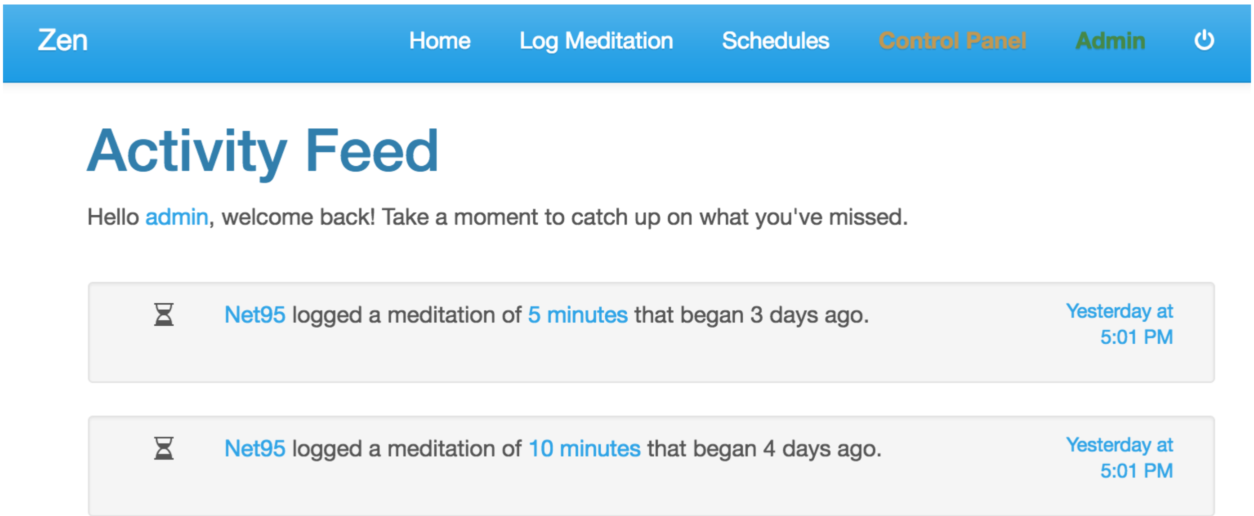


Figure 13: The study platform with a modified theme.

The configuration file, as mentioned above, contains parameters that specify the links in the navigation bar and the footer; these parameters allow the position of the link to be specified, the text to use, whether the link should appear for authenticated users and whether they should open a new browser window. Additionally, pages created by study administrators can be placed within the `views/pages` directory where they are parsed and made available. The configuration allows the URI slugs[13] and page titles to be specified but sensible default values are computed if these are omitted. An example configuration for adding a page explaining the rules of soccer can be found in Code Listing 2.

[23] This deployment can be found online (Study Platform Deployment with Alternative Skin, 2016).
[24] The git commits for the modification can be found on the alternative branch of the repository (lukem512/zen alternative branch, 2016).

Code Listing: JavaScript

```
soccer: {
  title: 'Rules of the Game',
  href: '/howto',
  guestOnly: true,
  footer: {
    display: true,
    position: 'right'
  }
}
```

Code Listing 2: The configuration options needed to add an additional page, visible only to guests and displayed in the footer and the navigation bar.

Content

The platform content was implemented using Jade templates that could display pledge information, schedules and fulfilments, and had corresponding forms for creating and editing them. Templates were populated via a database, queried using parameters specified within the URL. A template was also defined for displaying users; this page retrieved the recent activity of the user via an AJAX call to the API and displayed it in a feed.

The feed is constructed by querying the database for all schedules, pledges and fulfilments made by a particular user, or the users within a group, and humanising them by performing string processing. The data is formatted to use the correct tense, depending upon whether the action was performed in the past, will be performed in the future or is currently occurring, and so the parts of speech, such as ‘are’, ‘is’ and ‘they’, conform to one another correctly. Dates and durations are formatted by bucketing small times into ‘a few seconds’, intervals into minutes and, after an hour has elapsed, into hours. Some of the date and time formatting makes use of the popular *moment.js* library (Moment.js, 2016).

Schedules are displayed on a calendar, rendered using client-side JavaScript and CSS. The calendar makes use of the *fullcalendar* library (FullCalendar, 2016) and is populated using an API endpoint that provides JSON data in the expected format. The calendar was modified by implementing custom event handlers when navigating the calendar, switching the view and clicking on events, causing the user’s browser to navigate to a unique URL and allowing schedules to be bookmarked and shared. During the initial phase of implementation several JavaScript calendar libraries were trialled and a minor improvement (getX functions for Year, Month and Day - Pull Request #591 - Serhioromano/bootstrap-calendar, 2016) was submitted to an open-source library during that time.

To log an activity a participant can either do so retrospectively, by entering details of the start and end times into a form, or in real-time using a timer. The real-time feature communicates at regular intervals with the API using AJAX to retrieve information about the schedule or schedules pledged to by the participant, the online status of other pledged participants and to send updates to the platform about the duration of the fulfilment. The lifecycle of the real-time fulfilment involves sending an initial message to the API, creating a fulfilment flagged as `ongoing`, then sending periodic `keep-alive` messages with the fulfilment duration.

Once a real-time fulfilment has been completed a `finish` message is sent, removing the `ongoing` flag and marking the fulfilment as completed. If the participant cancels a real-time schedule a message is sent to the API causing the fulfilment to be deleted. This behaviour can be seen in Figure 14.

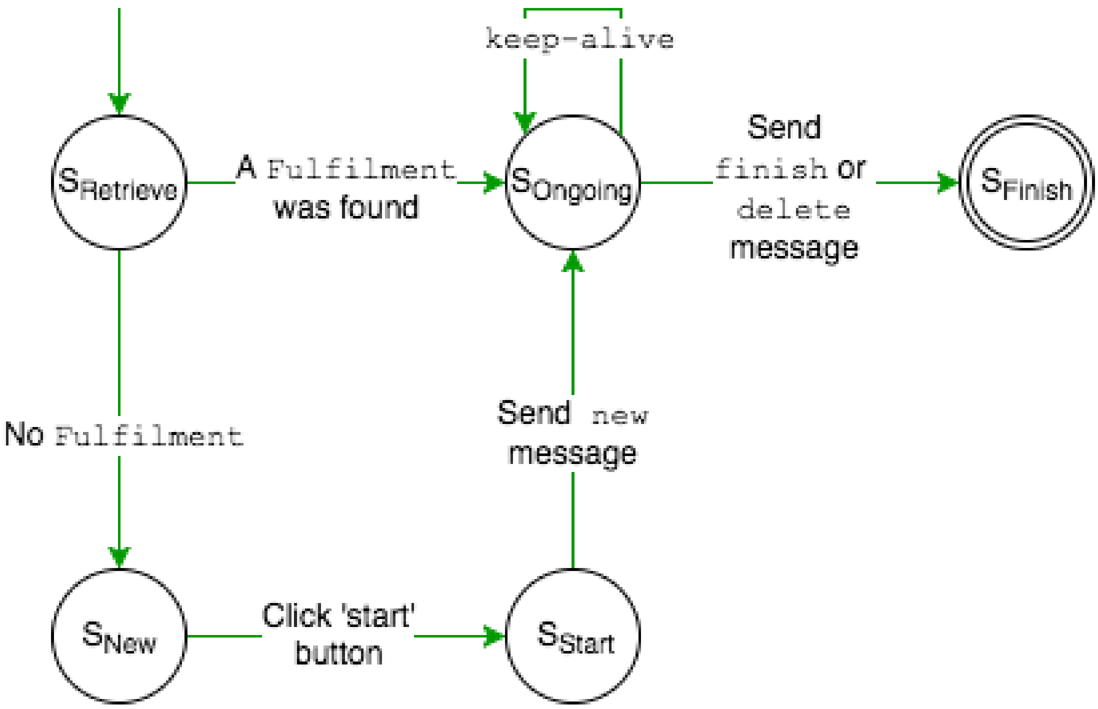


Figure 14: The state transition diagram for a participant logging a real-time activity.

The real-time feature is robust towards participants closing their browser window and opening multiple browser windows. The initial API message checks for ongoing fulfilments and, if one is detected, it is displayed and automatically resumed from the correct position. This behaviour occurs for 15 minutes after which time the fulfilment is automatically completed.

The platform makes use of HTML forms in many places, such as the creation and modification of schedules and fulfilments. Each form that is presented to the user, and wherever an AJAX request is made to the API, extensive client-side validation is performed using JavaScript. By performing the validation on the client-side the server-load is reduced, increasing the performance of the platform and reducing the overhead associated with transferring invalid requests. An additional advantage is that error messages can be quickly presented to the user without requiring a page reload, improving the responsiveness of the platform and improving the user experience. All validation is duplicated on the server for security but without the additional overhead of generating error messages and transferring these to the requesting page.

When validation fails a prominent error message is presented to the user. The error messages describe the problem and the required input. When selecting a value from the date and time picker form components simple validation is performed and, if a sensible suggestion is identified, the value is modified accordingly. An example of this behaviour is when selecting start and end times for a schedule: when a start time is specified an end time is automatically populated for 15 minutes ahead of it; changing the start time will update the end time until an end time is defined by the user.

Analysis Platform

The analysis platform was implemented within the administrator control panel as a collection of standalone routes. These routes are accessible by administrators but could, with simple modification, be exposed to other, theoretical classes of user. The platform is accessible from the control panel via a button or by accessing the `/admin/analysis` URI.

As per study criterion *5*, all code was to be kept entirely within Node.js, for simplicity and ease of deployment, ensuring that as a minimum number of technologies were employed. The option of using existing technologies such as Python with *SciPy*[25], which provided much of the requisite functionality, was explored, however these would add significant difficulty to deploying the platform and re-implementing the necessary methods seemed to be the better option.

Various descriptive statistics such as the mean, mode and median are computed for each user and can be viewed for specified groups. The range of values, including the minimum and maximum values, can also be viewed along with measures of spread: the variance and standard deviation.

The analysis platform provides the ability for a researcher to perform analysis of variance, or ANOVA. The analysis of variance is a parametric statistical test that partitions the samples into groups and analyses the variance of the groups, comparing it to the distribution of the original data and computing a score known as the *F-value*. The algorithm works by computing the sum of squares and degrees of freedom for the sample groups, then taking the ratio of these values to compute the mean square and the F-value. The formulae can be seen in Equation 1.

[25] SciPy is an open-source library of scientific functions and tools for Python.

$$SS_{Total}: SS_T + SS_R$$

$$SS_T: \sum_{i=0}^{N} \left((\mu_i - \underline{S})^2 * n_i \right)$$

$$SS_R: \sum_{i=0}^{N} \sum_{j=0}^{n_i} (s_j - \mu_i)^2$$

$$DF_{Total}: \sum_{i=0}^{|S|} (n_i)$$

$$DF_T: N - 1$$

$$DF_R: DF_{Total} - N$$

$$MS_x: \frac{SS_x}{DF_x}$$

$$F: \frac{MS_T}{MS_R}$$

Equation 1: The formulae for deriving the *F-value* of the ANOVA test, where *S* is the sample group, *N* is the number of sample groups and n_i is the number of observations in sample group i.

To interpret the F-value a p-value is computed; this is the probability that the differences in the sample variances could have occurred by chance and that the null hypothesis, that all samples are equal, is true. The p-value is computed by sampling 10,000 values from an F-distribution and finding the probability that a value is less than the F-value; this procedure is known as the F-test.

Several variations on ANOVA are used: Levene's test and the Brown-Forsythe test are the most common of these. The variations first transform the sample data before performing analysis. Levene's test takes the absolute difference of each observation and the sample mean whilst Brown-Forsythe takes the absolute difference of each observation and the sample median. A variant of Levene's test is also used where the squared difference, rather than the absolute difference, is taken. Typically, Brown-Forsythe is used in-place of ANOVA when it is suspected that sample data is not perfectly Gaussian[26], as it is more robust to non-normal data. These methods have been included as alternatives to ANOVA.

[26] Data is said to be Gaussian if it conforms to a Normal, or Gaussian, distribution.

The platform also offers nonparametric analysis by implementing the *Mann-Whitney U Test*[27]. The test works by ordering observations taken from two samples - in this case, the total time spent fulfilling an action by members of two groups - and ranks them according to an algorithm; the samples themselves are then ranked by taking a sum of the ranks of their observations and, from this, a *U value* is computed. The formulae can be seen in Equation 2.

$$R_s \colon \sum_{i=0}^{n_s} \{r_i \colon o_i \in s\}$$
$$U_s \colon R_s - \frac{n_s(n_s + 1)}{2}$$

Equation 2: The formulae for deriving the *U value* of the *Mann-Whitney U Test*, where r is the rank, o is the observation, n is the number of observations and s is the sample.

As mentioned, all analysis code was to be kept in JavaScript; the NPM package manager was searched for a module that would provide the relevant functionality and it was noticed that a suitable candidate did not exist[28]; as such, it was decided to create a new library. The test methods were all implemented separately, creating minimal, generic implementations that were packaged and made available via NPM. These implementations have been extensively tested, the method for which is discussed in the Testing section.

The packages are named *anova* (anova, 2016)**,** *brown-forsythe-test* (brown-forsythe-test, 2016), *levene-test* (levene-test, 2016) and *mann-whitney-utest* (mann-whitney-utest, 2016). A lightweight statistics module was also created and is named *statistical-methods* (statistical-methods, 2016). The modules for the Brown-Forsythe test and Levene's test are the only available implementations in JavaScript whilst the other implementations are not unique but are the smallest available. The modules have been downloaded over 1,000 times as of 26/04/2016, less than a month after publication.

Security

The security of the platform was a high-priority and considerable effort was spent ensuring that unauthorised access to the platform was impossible and that participant data was secure. Although the platform does not store any personal data about the participants, a decision made to aid anonymity, the protection of data gathered through a study is clearly important, both in terms of scientific rigour and in protecting the privacy of participants and the reputation of the research body.

Much of the functionality of the platform is exposed via an API that the webpages query using AJAX; this API handles user authentication, the retrieval of schedule, pledge and fulfilment data, the creation of new schedules and user administration. The API includes a

[27] This is also known as the *Mann-Whitney-Wilcoxon* or the *rank-sum test.*

[28] A library called *jsRegress* exists providing several statistical tests. The functionality of *jsRegress* surpassed my requirements, however, and the library had not been updated in 2 years, so it was decided to forgo it in lieu of creating a new library for increased control and a reduction in code size and dependence.

three-tier authentication system where some endpoints are exposed to guest users, some only to authenticated users and some to administrators.

The API authentication is performed using a *JSON web token* - a cryptographically secure string that identifies a user, passed via POST parameters or stored in cookies. Upon signing in a user is sent a web token; this is generated using a private key, `tokenSecret`, stored within the platform configuration file. The token contains their username and privilege level and accompanies all future API requests. The use of JSON web tokens is industry standard, conforming to the RFC 7519 open specification (RFC 7519 - JSON Web Token (JWT), 2015)**,** and allows two-way communication to occur in a fast and secure manner. The advantage of using web tokens is their speed and small size, allowing them to be passed as HTTP variables and transmitted quickly; and their self-containment, the data contained within a web token can be used to validate it (JSON Web Tokens, 2016).

User authentication is performed by entering a username and password on the sign in page. The password is hashed using *bcrypt*, a slow-hash resistant to brute-force attacks due to the relatively long time required to compute it[29], and compared to a hash stored in the database. Storing the hashed password rather than the plaintext ensures that the passwords are still secure should the database be compromised. A unique, random salt is generated for each user, further increasing the time required to crack the passwords. The authentication process can be seen in Figure 15.

[29] The use of slow-hashes for password storage is explained by Open Wall, the creators of the password cracker *John the Ripper* and several other security tools (Differences Between Fast Hashes and Slow Hashes, 2012).

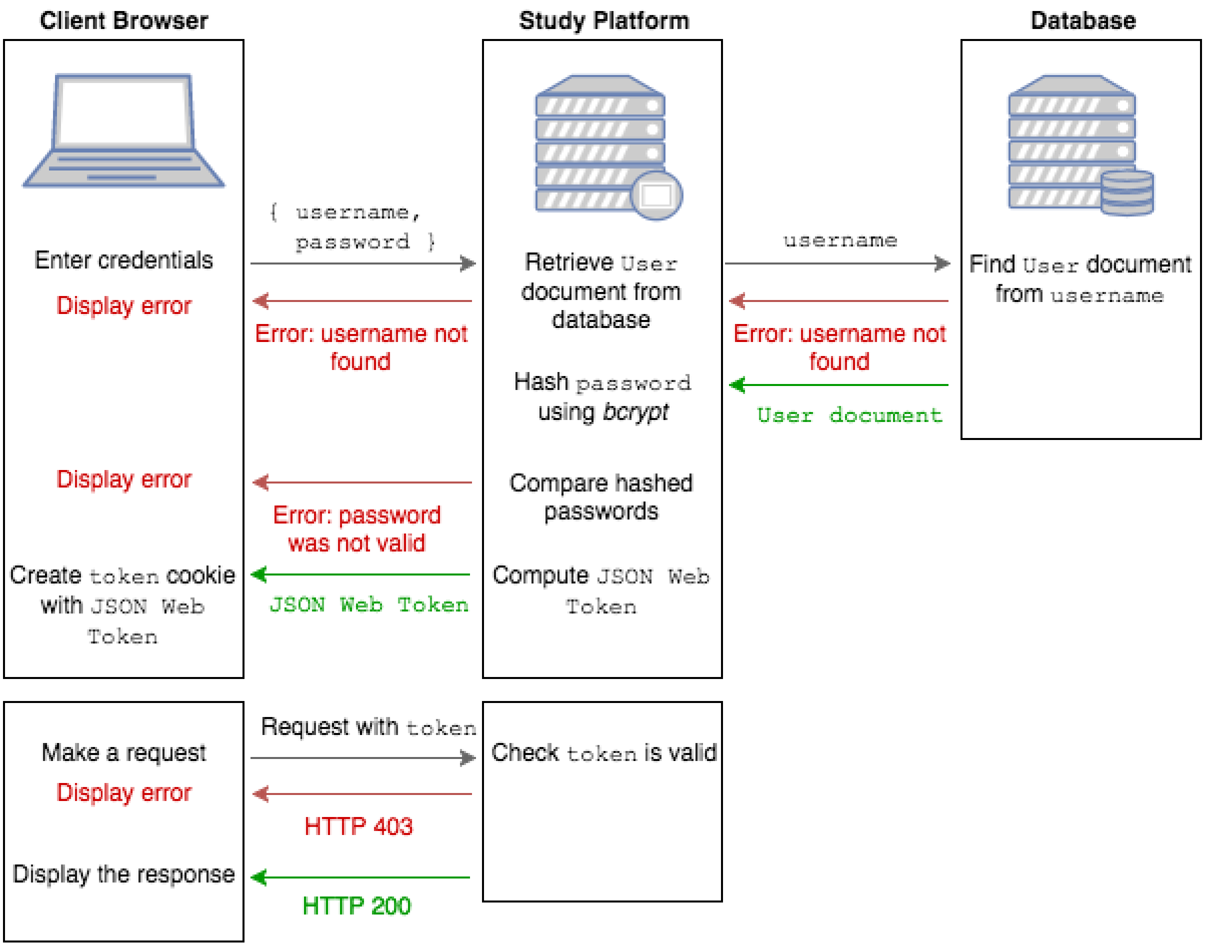


Figure 15: The programmatic flow of user authentication. Responses in red are errors whilst responses in green are successful. Requests are shown in grey.

The platform API performs authentication using custom middleware, written for Express.js. Middleware, in the context of the Express framework, is a term used describe a small, chainable function that sits between an HTTP request and the route handler, typically performing authentication, logging or transforming input. Middleware were defined for all tiers of authentication and for determining participants that are within specified groups or the creators of specified schedules, pledges and fulfilments, allowing these to be authorised appropriately.

The database used within the platform, MongoDB, is renowned for being secure due to its resilience to conventional injection attacks; however, a naïve install exposes several security holes (Kirkpatrick, 2013). Care has been taken to configure the database to prevent unauthorised access and to secure communications between it and the platform. All communications are encrypted using SSL, preventing eavesdropping and ensuring the integrity and the privacy of the data; this is important as all data is sent in plaintext (Security Checklist, 2016). The platform supports secure database connections using SSL or TLS by appending `?ssl=true` to the connection URI, specified in the platform configuration file, and configuring the database appropriately. As previously mentioned user passwords are stored in hashed form ensuring that, should a connection be initiated via an unsecured connection, the user account is not compromised. The database connection has also been configured to require a username and a password - these should be generated using a cryptographically secure random number generator or another secure method; by default, there are no credentials required to connect to a MongoDB instance.

When constructing queries using user input, such as a parameter from the URI of an API endpoint or from a form,, care has been taken to sanitise all database arguments, removing all modifier strings and $ characters, ensuring that NoSQL injection can not be performed[30]. By sanitising input in this manner it can be safely passed to the database without fear that content will be modified or arbitrary code executed.

Additionally, any arguments passed using the HTTP methods, such as being encoded in the URI, are parsed before display; this ensures that cross-site scripting (XSS) attacks cannot occur[31]. The prevention of XSS is particularly poignant for the authentication page which accepts a redirect argument[32]; here the argument is sanitised by removing any hosts that differ from the current page and any protocols that are not HTTP and HTTPS.

Finally, the platform is served over HTTPS, encrypting all communication between the client browser and the server. This is a requirement for any secure application as it prevents man-in-the-middle, DNS spoofing and redirection attacks[33] from being performed on unsecure networks.

Performance

After implementation the study platform was benchmarked to gain insight into the maximum number of concurrent participants it could handle. The benchmarks were performed using a standard tool, *Apache Bench*, and an increasing number of concurrent requests were made to an instance of the study platform, remotely hosted on the Heroku free tier. The benchmark used 1000 requests in total and these were made concurrently. The response times and the dropped-packet rate for each were recorded.

The tests revealed that no packets were dropped, even with 250 concurrent requests, and that the response time increased by an order of magnitude from around 300 ms to around 3300 ms. Figure 16 shows that the decrease in performance was linear.

---

[30] NoSQL injection is an attack whereby a malicious user enters input containing database commands or modifiers that, when passed to the database in a query, will be executed. Due to the construction of MongoDB queries this is less of a problem than the similar SQL injection attacks, performed on MySQL, PostgreSQL and other relational database technologies.

[31] A cross-site scripting attack is where malicious client-side code, such as JavaScript, is executed by tricking a vulnerable page into displaying it for the user.

[32] This is a URI is in the form of `/auth?r=http://zen.lukemitchell.co.uk/meditation.`

[33] These are attacks that modify the results of requests for web pages and DNS addresses, returning malicious results designed to steal credentials or alter user behaviour.

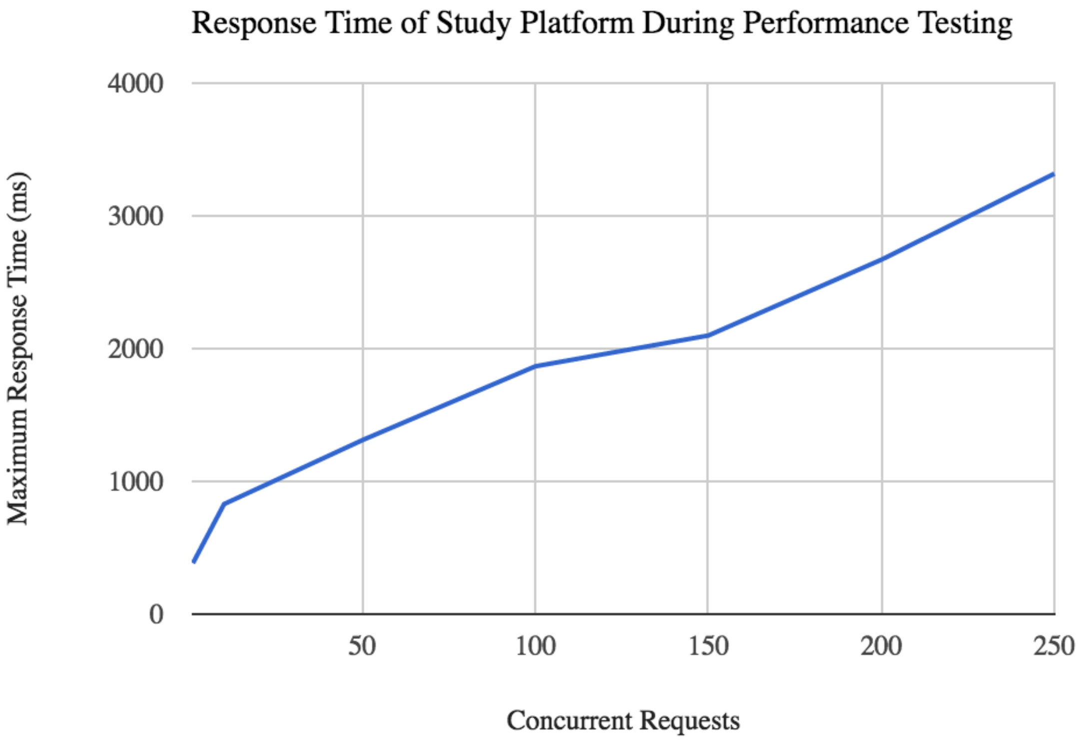


Figure 16: The maximum response time of an increasing number of concurrent requests made to a remotely hosted instance of the study platform.

Testing was not performed for greater than 250 concurrent connections due to hardware restrictions, however, it is probable that this trend would continue. A load of 250 concurrent requests is equivalent to 100 users simultaneously logging activities in real-time, or navigating their calendar. The performance at this level of concurrency, whilst fairly slow, is not unusable and could be tolerated.

Additionally, due to the single-threaded nature of application and the strong de-coupling between the front-end and the database, the study platform can be horizontally scaled with ease. Multiple instances of the software can be run, in parallel, without requiring additional locking mechanisms to be implemented, giving a roughly linear increase in performance that can be used to match the performance decrease of additional users. If a desired performance level is required, for example a 1000 ms load time for 100 concurrent users, the number of instances can be scaled to achieve this: in this example 3 instances would be required.

Development Process

When implementing the platforms good development process was kept wherever possible; this included keeping detailed bug reports, writing descriptive commit messages and employing language best-practises. This is particularly important, given the intended public use of the software, as the difficulty in gaining familiarity with code is greatly reduced when it is consistent, neat and uses common conventions. The bug reports and commit messages will be available hereafter in the Git repository.

All code is well-commented, consistently-indented and conforms to language best-practices (JavaScript Standard Style, 2016) resulting in good readability and ensuring it is easy to maintain. Any subroutine that has been used in multiple locations is available as a function to avoid code duplication and reduce the time taken to rewrite or update it. The code conventions used for the project can be seen in Table 3.

Convention	**Description**
Use single quotes when defining strings	JavaScript accepts both single and double quotes as valid. Care has been taken to use only single quotes when defining strings.
Escape all double quotes when defining strings	JavaScript accepts *unescaped* double quotes when defining strings with single quotes. This is messy and can lead to difficult to identify bugs if double quotes are inadvertently used, however, so this has been avoided.
Indent all code by two spaces	-
All names are *camelcase*[34]	All variables and functions are named using camelcase. This is with the exception of MongoDB models where the convention is to capitalise the first letter and use the singular.
Prefix identical names with an underscore for inner functions	When a variable or function is named identically the copy in the innermost function, that is, at the top of the call stack, uses the name prefixed with an underscore.
Always use curly braces with conditional or functional statements	JavaScript accepts single-statement conditional and function statements to omit these. This can lead to difficult to identify bugs[35] and has been avoided.
Always follow control statements with a space, preceding the opening curly brace	-
Always use a line break before an else statement following the closing curly brace	-

[34] Camelcasing involves capitalising the first character of every word except the first word and joining them without space.
[35] An example of this is the notorious Apple SSL bug “goto fail (Anatomy of a “goto fail” – Apple’s SSL bug explained, plus an unofficial patch for OS X!, 2014).

Always follow keywords such as `if`, `switch` and `do` with a space	-
Always follow a closing parenthesis with a space if its preceding a curly brace	-
Never follow an opening parenthesis or precede a closing parenthesis with a space	-
Always follow a statement with a semicolon	JavaScript accepts statements without a semicolon as long as whitespace is used instead. This leads to messy code and has been avoided.
When using the `onReady` handler use the jQuery `$(function() { /* .. */ });` notation	This handler can be expressed in several ways and this convention has been included for consistency.
All dead code[36] should be removed	The exception is when code is left in and explained or psuedo-code is given.
Lines should be less than 80 character long	The exception is when formatting a string and following this convention will lead to unreadable code.
Broken lines should be indented at the beginning	-

Table 3: Code conventions employed when implementing the software.

The technique that was employed during development was an iterative approach with regular user testing. This approach was adopted as many of the design goals focussed on simplicity of use and availability to a wide range of participants; as such it was imperative that users began testing the platform and providing feedback as quickly as possible. The user testing is discussed in more detail in the Testing section.

I relied heavily upon the Git source-control tool whilst implementing the platform, using the *GitHub* service to host a remote repository, track issues and write documentation. The use of source-control is the standard-protocol for many open-source projects, particularly when hosting with GitHub, and the tool simplifies collaboration by combining the features of Git with referencing and the ability to comment. A master branch was maintained for the production-ready code and additional branches were used when working on a specific feature area, such as the creation of the feed or the analysis platform. This kept similar code together and improved the readability of the commit messages, particularly when they were squashed into a single message of the form “Added duplicate schedule button; fixes #20”.

[36] Dead code is commented-out code that serves no illustrative purpose and has been left in from a previous version of the software.

Issues were tracked using the GitHub issue-tracker and care was taken to reference these in commit messages when they were resolved. This is an industry-standard practice that reduces can significantly reduce the effort required to track down and fix bugs that resurface, as well as providing accountability when used in a multiple-user environment. Throughout the project 29 issues were reported, of which 10 remain unresolved.

Two builds were maintained when developing and testing the platform: a production build, pushed from the master branch and used by the study participants; and a staging build, containing additional fixes, functionality and modifications. The two builds were hosted separately and were isolated to ensure that any issues introduced did not affect the study. This is discussed in more detail in the Testing section. This is equivalent to using the *maven* build manager for Java or *apt* for *Debian* and *Ubuntu.*

To manage project dependencies NPM was used, a package manager for Node.js, and Bower, a package manager for client-side JavaScript. Both of these package managers have configuration files and all dependent technology has been specified, along with the requisite versions. Where possible small packages were re-implemented or packages without many dependencies were used, aiming to avoid *dependency hell* and both bloat the software and increase the complexity of installing it.

Automated build tasks using Grunt were also employed: the platform installs the relevant dependencies using NPM and Bower, *minifies*[37] the client-side stylesheets and scripts using Grunt and then moves them into the public directory where they can be served to the user. By performing minification the amount of data transferred between the platform and user is reduced, increasing the load-time of pages and reducing the running costs of the platform. The build tasks also streamlines the process of installation by reducing it to a single command "`npm install`". By using an automated build script, the install process has been kept simple, as per design criterion 14**.**

Future Work

There are several key improvements that could be made to the study and analysis platforms to broaden the range of studies that can be performed using the software. These improvements are summarised below:

- Refactor the study platform to allow multiple endeavour types
- Implement the Robust Rank-Order nonparametric statistical test

The first of these improvements, the inclusion of multiple endeavour types, is a further generalisation of the study platform to include four endeavour types: a collective goal, in which a common goal is set to be achieved by everybody on the platform, working individually, where the actions of all participants contribute towards the shared goal; a collective practice, in which a common goal is set to be achieved by everybody on the platform, working together, where all participants take part in a shared activity to contribute towards the shared goal; an individual, homogenous goal, in which individuals have their

[37] Minification is the process of condensing code by removing superfluous whitespace and characters and, often, by renaming variables to shorter names.

own goals, all relating to the same task, which are visible to others on the platform; and, finally, an individual, heterogeneous goal, in which individuals have their own goals, all relating to different tasks, which are visible to others on the platform. Currently the platform only allows individual, homogenous goals to be performed; however, by generalising the method for scheduling and pledging to schedules, and by allowing different cohorts to be assigned a different endeavour type, the effects of social pledges in many more situations could be explored.

The second of these improvements involves the implementation of an additional nonparametric statistical test, the Robust Rank-Order. This test is more suited to medium-sized samples as it reduces the false-positive rate in comparison to the Mann-Whitney U test (Feltovich, Nonparametric differences in Wilcoxon-Mann-Whitney and Robust Rank-Order Tests., 2003) and, additionally, a table of critical values for a much larger range of p-values exists, allowing conclusions about less-likely significances to be drawn.

Additionally, there is an improvement that could be made to the security of the platform: that password-hashing is performed client-side, in the browser of participants. The hashing algorithm used is a slow-hash, designed to be resistant to brute-forcing. Whilst this is beneficial as the platform stands, as it protects password integrity from database leaks or hacks, it exposes a weakness to denial-of-service attacks. A denial-of-service attack can be performed by submitting a large number of authentication requests with dummy passwords; each time a request is received the server has to perform an expensive operation to hash the password. This can be mitigated by performing the hash on the client-side and, additionally, by placing a limit on the number of requests that are processed by each connection in a given time period.

During the interview process I was made aware of two main problems that participants encountered: a lack of guidance on how to meditate and a lack of integration with existing calendar tools, such as those present on smartphones. The desire for integration with a smartphone seemed particularly prominent as one participant commented “I would have liked there to be an alarm or reminder system in place once you had scheduled, or a way to link it to your phone’s calendar so there was a way of being reminded” and another said “if it could be linked to a phone or could be created as an app which could send you reminders with encouraging messages it would help.”

Implementing a function to export the schedules to common calendars, such as those found on *Android* and *iOS* smartphones, would satisfy these requirements and would require a minimum of work. A more complete solution would be to fully integrate the platform with Google calendar so that the import process would only have to be performed once by participants. The Google Calendar format can be viewed on both Android and iOS, as well as on the Google website, and so this would be accessible to all participants and condense their workload to a single step.

The provision of guidance on how to meditate is a shortcoming of the study design and a problem that could be prevented by writing a short document on the task being studied, or by recruiting participants with prior knowledge about the task. This requires no modification to the study platform.

There are also several improvements that can be made to the software development process. These improvements are not essential to the function of the platform, however, by implementing them they allow for increased readability and reduce the work required to modify the platform in the future, if it is necessary to do so. These improvements are as follows:

- Implement a full suite of unit tests for the study and analysis platform. At present the statistical components of the analysis platform have been unit-tested but much of the study platform has not
- Utilise the *babel* compiler and convert the platform code to the latest version of JavaScript, ECMAScript 2015
- Utilise *jslint* to formalise the code conventions, described in the Development Process section, and to establish coverage tests based upon this
- Utilise GitHub's commit-hook feature to perform continuous integration, such as running the coverage tests automatically and reporting failed builds

Despite these improvements the platform received a large amount of positive feedback during qualitative feedback, performed after the study had concluded. One participant commented "I liked the look of it, it was intuitive and easy to use", another said "Yes, it was user friendly, very simple to use and looked very sleek." and yet another said "A pleasant and easy to use system that made me conscious of how much I was meditating - an awareness that helped me meditate more freqeuntly [sic]."

Testing

This section describes the test procedure undertaken for the implementation and further development of the study platform, the analysis platform and several open-source packages that were developed as peripheral software. Two varieties of tests are performed: verification via unit testing and validation via user testing. Verification is achieved by defining unit tests in which an input, such as a sample fulfilment, is processed and the results compared to a known, correct set; these tests are performed automatically to ensure changes to the software do not break existing functionality. Validation is achieved by asking focus groups and prospective participants to use the platform and provide feedback; this ensures the software is usable for the intended task and fulfils the design criteria.

As mentioned in the Specification section, I implemented both the study and analysis platforms using the latest HTML5 and CSS3 standards. To ensure that the software conformed to these standards the *W3C validator* ((X)HTML Validation results for zen.lukemitchell.co.uk, 2016) was run periodically, and any problems fixed. The *jshint* package was also used: jshint is an NPM module that was integrated into the build process to provide warnings about violations of the JavaScript conventions used in the platform code.

When designing the study platform several user scenarios were prototyped on paper and these were shown to prospective participants, in focus groups, to gauge their feedback. Questions were asked about their understanding of the user interface, such as "if you wanted to schedule a meditation practice, how would you expect to do it?" and "if you wanted to change your password, where would you look?". This feedback was invaluable and several modifications to the initial designs were made as a consequence; these changes included adding buttons to the calendar for switching between different views (day, week and month) and displaying schedules that have been pledged-to in green.

Once implementation of the study platform had begun a second round of user testing was performed. Many of the same users were recruited, alongside an additional 20 users gathered from social media and the author's extended family, to complete a list of tasks and provide feedback. The tasks included the following items:

- Logging into the platform with pre-generated credentials
- Creating and modifying several schedules
- Pledging to join the schedule of another participant
- Changing their password

The final task was to find the author's contact details from the platform, which were located in the Contact Us page, and to email their feedback to the specified address. The full list of tasks has been included in Appendix A.

This period of testing was also extremely valuable as it facilitated the discovery of several bugs with earlier versions of Internet Explorer, including the absence of the placeholder attribute, and gave suggestions for the behaviour of various forms, such as choosing a sensible time to suggest as the start time when creating a schedule, using increments of 5 minutes and automatically populating the end date with the selected start date.

During the implementation of the analysis platform 5 NPM modules were created. Each of these modules was developed using test-driven development techniques; the implementation process was begun by defining tests derived from data found online. A simple test framework was created, comprising of an input payload and an expected output and used this to test the modules. The sample data used in the tests was sourced from academic resources such as a revision sheet for Plymouth University (Analysis of Variance (ANOVA), 2016).

The NPM modules are continuously tested using a tool called *Travis-CI*: this performs a test command, specified in the `package.json` file present in all NPM modules, each time a change is pushed to the GitHub repository. The tool used GitHub *WebHooks* to do this and generates a small image, known as a badge, displaying the test results that is included in the `README` on the repository page.

During the study period two changes were deployed to the platform: a fix for the locale setting which failed to respond to the beginning of British Summer Time and the inclusion of buttons to duplicate and fulfil schedules, as requested by participants. To ensure that these changes would not break the platform they were first deployed to a staging environment (Zen (Staging), 2016). The staging environment is identical to the production environment except that it uses an isolated database, allowing tests to be performed without damaging the integrity of the data stored on the study platform. Additionally, if any bugs were introduced, these could be identified without inconveniencing the participants.

Before the study began, and before migrating staged changes to the production environment, a list of tests was manually performed. This was a checklist designed to explore all aspects of the user interface and platform functionality and was designed to mimic typical participant behaviour. The checklist can be seen in Table 4.

Test	Description
Log into the platform	-
Create a Schedule	-
Edit a Schedule	-
Delete a Schedule	-
Pledge to a Schedule	-
Unpledge to a Schedule	-
Duplicate a Schedule	-
Multiple participants pledge to a Schedule	This ensures that the display is correct.
Multiple participants unpledge to a Schedule	This ensures that the display is correct.
Edit a Schedule that has been pledged to	This ensures that participant pledges are updated.
Delete a Schedule that has been pledged to	This ensures that participant pledges are also deleted.
Attempt to create a Schedule in the past	This is not allowed.
Attempt to remove a Schedule in the past	Once a Schedule has passed it cannot be deleted.
Log an action retrospectively	-
Log an action in real-time	-
Incorrectly finish logging an action in real-time and resume it	The real-time functionality is robust to participants opening multiple browser windows and exiting prematurely. The session should resume from the correct time if it is restored in 15 minutes.
Log an action retrospectively that fulfils a pledge	-
Log an action in real-time that fulfils a pledge	-
Log an action in real-time with another participant present	The other participant should be visible in the display.

Log an action in real-time with another participant absent	The other participant should be visible but obviously offline in the display.
Log an action that partially fulfils a pledge	The pledge should be shown as partially fulfilled.
Attempt to log an action in the future	This is not allowed.
Miss a pledge	This should be visible on the Schedule page.
Miss a pledge along with another participant	This should be visible on the Schedule page.
Miss a pledge that another participant has fulfilled	This should be visible on the Schedule page.
Miss a pledge that another participant has partially fulfilled	This should be visible on the Schedule page.
Delete a recent fulfilment	-
Edit a recent fulfilment	-
Attempt to delete an older fulfilment	Fulfilments over 15 minutes old cannot be deleted.
Attempt to edit an older fulfilment	Fulfilments over 15 minutes old cannot be modified.
Attempt to edit a fulfilment made in real-time	This is not allowed as the study has recorded the time.
Change the participant’s password	-
Log out of the platform	-
Log into the platform using the new password	-
Ensure that participants in a group cannot see ungrouped participants	This is important as participants in a group should not be aware of the activity of other groups.
Ensure that ungrouped participants cannot see participants in a group	This is important as ungrouped participants should be isolated.
Ensure that participants in a group cannot see participants in another group	This is important as participants in a group should not be aware of the activity of other groups.

Table 4: Tests to manually perform before migrating from the staging to production environment.

Documentation

Full documentation for the platform has been written and made available via the Wiki on the project GitHub repository (lukem512/zen - GitHub, 2016) The documentation contains typical user scenarios that can be customised and distributed to participants for future studies. Some example scenarios are the creation of a schedule, logging an activity as it is performed, logging an activity in retrospect and changing a participant's password. These pieces of documentation have all been written in a simple and user-friendly manner, including example screenshots to illustrate the requisite steps; the documentation is also written using the generic terms "schedule", "action" and "fulfilment", all of which can be customised as-per the platform dictionary.

The documentation also includes guides for study administrators; these guides describe how to customise the platform, how to deploy the platform locally and via Heroku, how to create users and groups for participants and how to perform analysis of the gathered data. The platform documentation, as with the software itself, is available under the *GNU* GPL open-source license.

Summary of Implementation

This concludes the implementation section. This dissertation required the implementation of 2 platforms: a study platform, facilitating the participants of a study to schedule and log their activity in a social manner; and an analysis platform, facilitating the analysis of participant data, gathered from the study platform. These platforms were implemented using current, industry-standard tools and techniques and have been created in a general, highly-customisable manner. In addition, several peripheral software components have been created and released under open-source licenses. A summary of the implementation is as follows:

- Implementation of a generic study platform, using Node.js, Jade, MongoDB and Express.js
- Implementation of an analysis platform, integrated into the study platform
- The study platform is highly configurable via a simple configuration file and example page templates
- The implementations of the above are secure and resistant to XSS, injection and man-in-the-middle attacks
- The implementations of the above are performant and horizontally-scalable with ease
- The implementations of the above conform to specified design criteria
- The use and deployment of the above is well documented
- The implementations of the above are thoroughly tested for verification and validation using users and unit tests
- The implementation of 5 NPM packages for statistical methods

Study

Design

To begin to answer some of the open research questions in social pledging a study was designed to investigate the effects of socially pledging an intention to fulfil a task upon the motivation of participants to perform it. The study involves using an instantiated version of the study platform, described in detail in the previous section, in which participants recorded their meditation practices. The study involved comparing participants that were aware of the pledges of other participants, creating a collective social norm, and participants that were isolated and only aware of their own activity. Meditation was chosen as the task to perform as it is non-competitive, thus well-suited to normification allows participants to practise alone and in any location. Participants can also meditate easily whilst at a computer or with a mobile phone allowing the real-time functionality to be used without distraction.

The hypothesis of the study is stated as follows: “given a regular task to perform, those that socially pledge their intention to perform it, at a specified time and date, will do so for longer, and more reliably, than those that do not socially pledge.” A secondary hypothesis is that: “given a regular task to perform, those that can see the social pledges of other participants will attempt to match their activity, leading to reduced variance with respect to a group that is not aware of the social pledges of other participants.” These hypotheses are derived from the finding of Chen et al (Chen, 2010) who found that a ‘boomerang effect’ is observed when participants are aware of the performance of others, and that participants typically perform closer to the mean; and Schultz et al, who found that manipulating the perceived descriptive norms results in increased performance.

The independent variable studied is the ability to view and interact with other participants, manifested through an ability to see the activity of and the ability to pledge to schedules created by other participants. The dependent variable is the time spent meditating by each participant, the average and variance of which will be statistically studied. The test will be conducted *between-subjects* as the independent variable will not be varied within each cohort; this allows statistical measures such as analysis of variance to be simply performed.

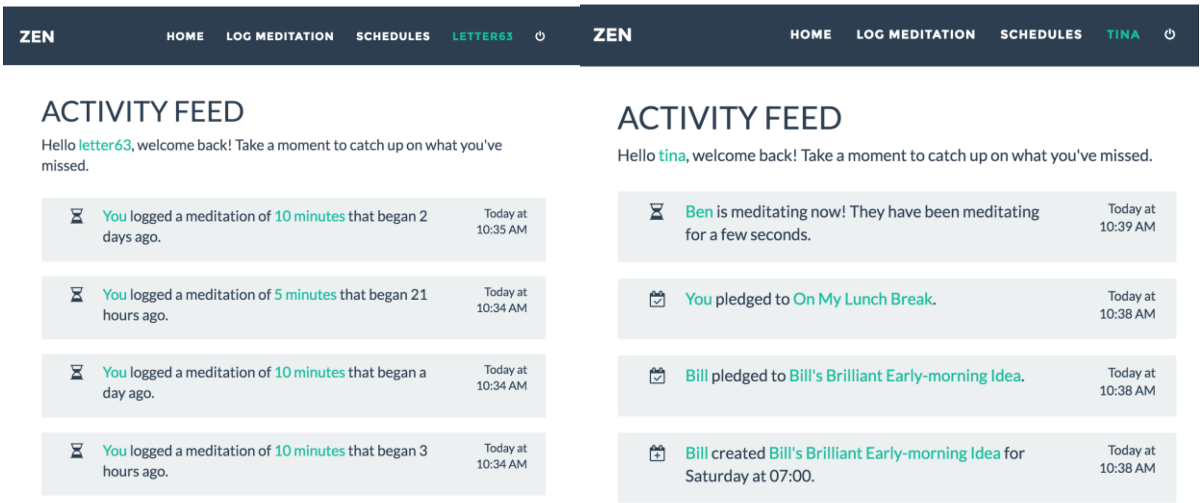


Figure 17: The platform as seen by participants in the CONTROL cohort (left) and the SOCIAL cohort (right).

Participants were told to use the online platform to schedule meditation sessions as they seemed appropriate, and to record any meditation sessions that they performed, either as part of a schedule of otherwise. The study platform was used to record information about the sessions recorded including their start and end times, whether they were scheduled, or pledged, whether the pledge was completely fulfilled and whether the session was recorded in real time or in retrospect. After the study had concluded qualitative data was gathered by conducting interviews and questionnaires with the participants. This was performed to gain insight into the motivations felt by the participants and to understand the patterns of behaviour visible in the data.

Experimental Method

Recruitment Process

The study involved recruiting participants into two cohorts: a CONTROL group, containing isolated participants, and a SOCIAL group, in which participants were aware of the activity of others. Each cohort was required to contain a minimum of 20 people as this quantity struck a compromise between being feasibly recruitable in the limited time available and providing enough data to yield a statistically significant result. Additionally, in previous studies, mentioned in the Technical Background, cohort sizes of around 20 were used (Preist, 2014)**.** The demographic for participants aimed to be as diverse as possible, ensuring that the results inferred from the sample were generalisable to the wider population. To achieve this a combination of social media, printed flyers and word-of-mouth to recruit participants were used.

A total of 48 participants were recruited for the study. Participants were then randomly shuffled into 2 groups of 24, ensuring random assignment and avoiding bias. Participants that never used the platform were considered drop-outs and were excluded from analysis, although queried as to why they failed to participate. The final cohort sizes were 11 participants in the SOCIAL cohort and 14 participants in the CONTROL cohort.

Most of the participants were gathered using social media and by word-of-mouth: a recruitment message was posted to the author's Facebook profile, encouraging their friends to do the same; a message was also posted to the author's Twitter profile as this has a large number of diverse followers[38]; an appeal was made to several prominent meditation-based Twitter users to *retweet* the recruitment message, reaching approximately 60,000 users; also used were emails, gaining additional help from the project supervisor to target relevant mailing lists. In total an estimated 70,000 people were reached, most of whom were via Twitter, Facebook and through direct emails.

The study included several requirements for participants: that they did not previously meditate for more than 60 minutes each week, that they had regular access to the Internet and that they were competent computer users. The foremost criterion was chosen to ensure a roughly equal level of initial motivation as it was believed that, by meditating for longer than an hour each week, a participant would have an established schedule and their own motivations for completing it. The latter criteria were chosen for practicality as the platform was hosted online and, if a participant was not competent at using a computer, they may face difficulties in using that platform that would result in an unfair conclusion being drawn from their activity or lack thereof.

Due to the study requirements an inherent bias was introduced into the participant demographic: participants that were did not have regular access to the Internet, were not competent computer users or existing meditators were not represented. Whilst this bias is undesirable it does not exclude a large proportion of the overall population as 78% of the UK population have *almost daily* access to the Internet (Internet Access - Households and Individuals, 2015); the level of computer competency is harder to measure, however, by assuming that regular users of social media are sufficiently competent to participate, this requirement includes 61% of the population**.** Due to the nature of the study these requirements were unavoidable and so care must be taken to only generalise as far as the population represented by the sample. Additionally, given the field of research involves digital communities in specific, this does not present a problem.

Many of the participants gathered through Facebook or flyering were university students in their early 20's and, as such, this demographic is considerable over-represented in the study. It is difficult to gauge the full demographic represented, however, as the study did not gather personally identifiable data; this was only gathered via the questionnaires and when conducting interviews and is not representative of the entire participant sample. Of the 12 participants interviewed, 6 were students and 8 were aged between 22 and 24. The non-student participants identified as working for human resources, as a CEO for a charity, as a music teacher, as a further education manager and as a barista, representing a wide range of employment.

[38] The Twitter account (Luke (@LukeCrypto), 2016) has 1,850 followers and are mostly interested in cryptocurrency; they are predominantly male and middle-aged but, due to an additional interest in ecology and sustainability, there are a smaller but still significant number of younger followers of all genders.

Qualitative Data

Once the meditation period, lasting 4 weeks, had been concluded, all participants were sent a qualitative questionnaire about their experience in an email. The questionnaire contained either 9 or 17 questions depending on which cohort it was sent to; the questionnaire for the SOCIAL cohort contained additional questions on their interactions with other participants. The questions were a mixture of *Likert scales* assessing the motivation of participants to meditate in a variety of situations, such as when they had scheduled a meditation or when they had pledged to join another participant's schedule, and open questions asking about positive and negative experiences. The questionnaire is attached in Appendix B.

In addition to questionnaire responses, qualitative feedback was gathered by conducting interviews with several participants from each cohort. The interviews were conducted with participants that were identified as 'light', 'average' and 'heavy' meditators, corresponding to the amount of time logged using the platform. The participants identified as desirable for interview, a subset of each cohort, were approached via email and asked for comments; 4 participants from the SOCIAL cohort and 2 participants from the CONTROL cohort agreed to be interviewed. The interviews were conducted in person or via telephone and were not recorded.

Results

The final cohort sizes, as discussed in the Experimental Method section, were 11 participants for the SOCIAL cohort and 14 participants for the CONTROL cohort. The drop-out participants were contacted to gain insight into their reason for not participating and the responses given were that they were "too busy" or, in one case, unable to due to illness. A response was not given by some participants and so it is not clear whether the reasons differed for participants in different cohorts; however, one participant from the CONTROL cohort, that logged a single meditation, commented saying that a negative experience they encountered was "feeling stressed when I couldn't find the time to meditate" and this feeling may have discouraged others from logging anything.

Quantitative Results

During the study 148 meditation fulfilments were logged using the platform: participants in the SOCIAL cohort logged 99 fulfilments whilst participants in the CONTROL cohort logged 49. There were also 102 schedules created by participants. This data can be seen in Table 5 and the distribution of fulfilments can be seen in Figure 18. The distributions are shown as 'meditation events' on the x-axis, representing the date and time, with duration shown on the y-axis. The colour of the meditation event identifies the participant that logged it and the shape identifies their cohort: events from the SOCIAL cohort are shown as circles whilst events from the CONTROL cohort are shown as squares.

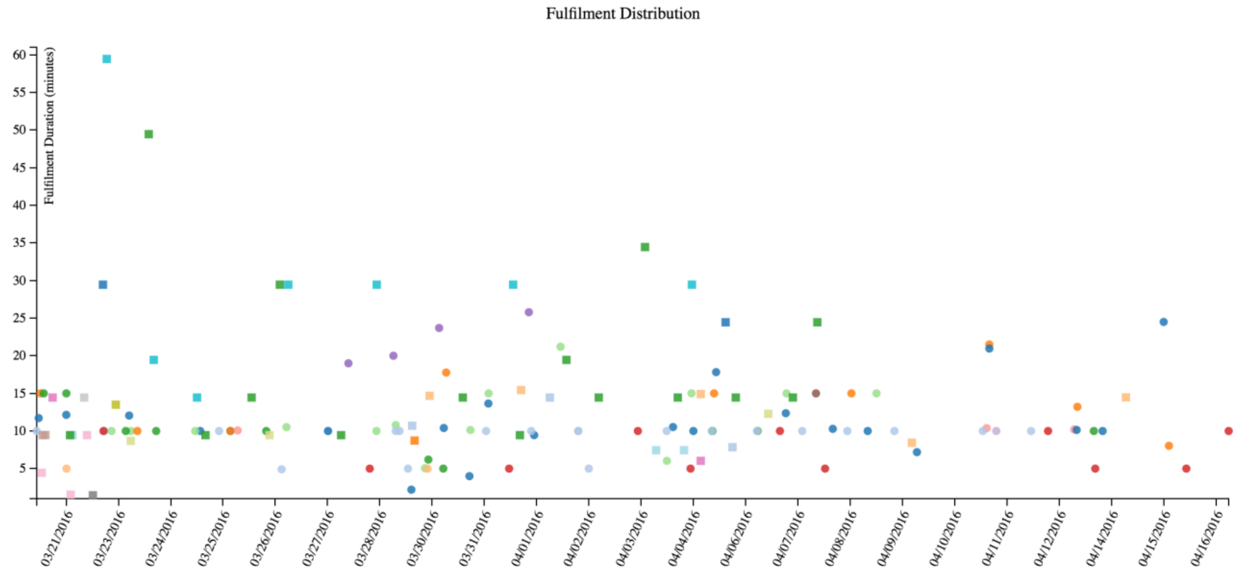


Figure 18: The fulfilments logged during the study by participants. Participants from the SOCIAL cohort are shown as circles whilst participants from the CONTROL cohort are shown as squares.

Cohort	Total Num. of Schedules	Mean Num. of Schedules	Total Time Scheduled	Mean Time Scheduled	Total Num. of Fulfilments	Mean Num. of Fulfilments	Total Time Fulfilled	Mean Time Fulfilled
SOCIAL	56	2.333	740	31	99	9	1079	98
CONTROL	26	1.04	645	26	49	3.5	822	58

Table 5: The number of schedules and fulfilments made by participants in each cohort. The unit is minutes, for times, and a count for other fields.

The total number of fulfilments logged by each participant can be seen in Figure 19. This indicates that participants from the SOCIAL cohort typically logged many more meditation fulfilments, as 4 of the top 5 participants, and 7 of the top 10, were from this cohort. This agrees with the averages in Table 5 as these show an average of 9 fulfilments per participant in the SOCIAL cohort and just 3.5 fulfilments per participant in the CONTROL cohort.

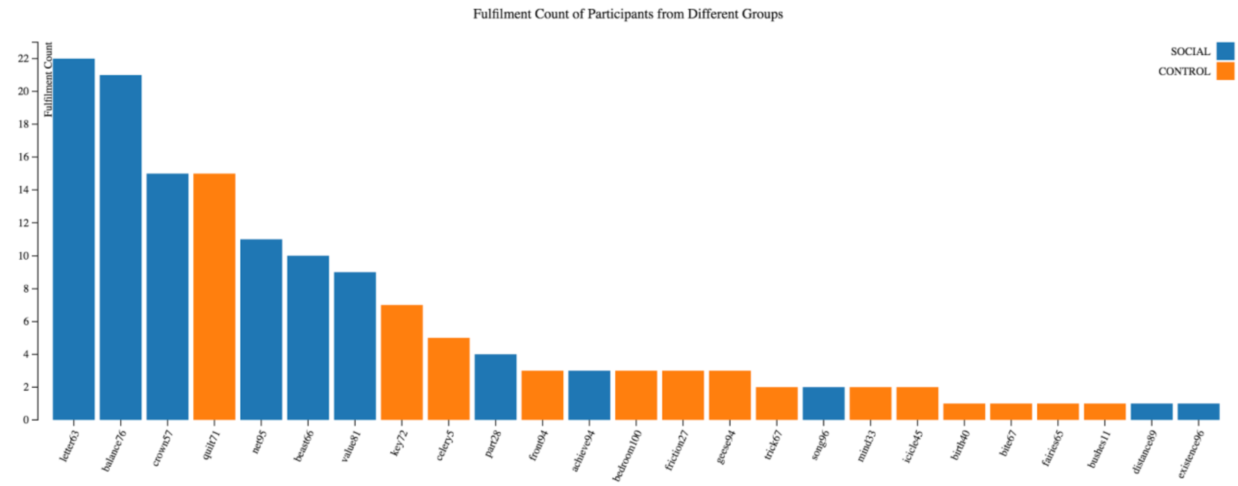


Figure 19: The total fulfilment counts of participants from both cohorts, shown in descending order.

The distribution of fulfilments across both cohorts, and for individual participants, can be seen in Figure 20. This seems to show that participants in the SOCIAL cohort meditated more regularly and, typically, for less varied durations. The descriptive statistics for fulfilments, shown in Table 6, is consistent with this: showing that the spread of fulfilment times for participants in the SOCIAL cohort is significantly less than that for participants in the CONTROL cohort and that the range is also much smaller, indicating that there are no outliers or exceptions to this.

Figure 20 indicates that most of the participants in the CONTROL cohort ceased meditating after 3 weeks, a week before the study was concluded, and that 5 participants from the cohort ceased meditating after 3 days. The participants in the SOCIAL cohort, on the other hand, logged meditations across the entire study duration. The figure also shows that most of the participants in the CONTROL cohort that logged a single meditation did so during this time, with the only exception being one participant, *bushes11*, that logged their sole meditation in the second week. Interestingly, the participants in the SOCIAL cohort that only logged a single meditation, of which there were half the number, did so much later in the study.

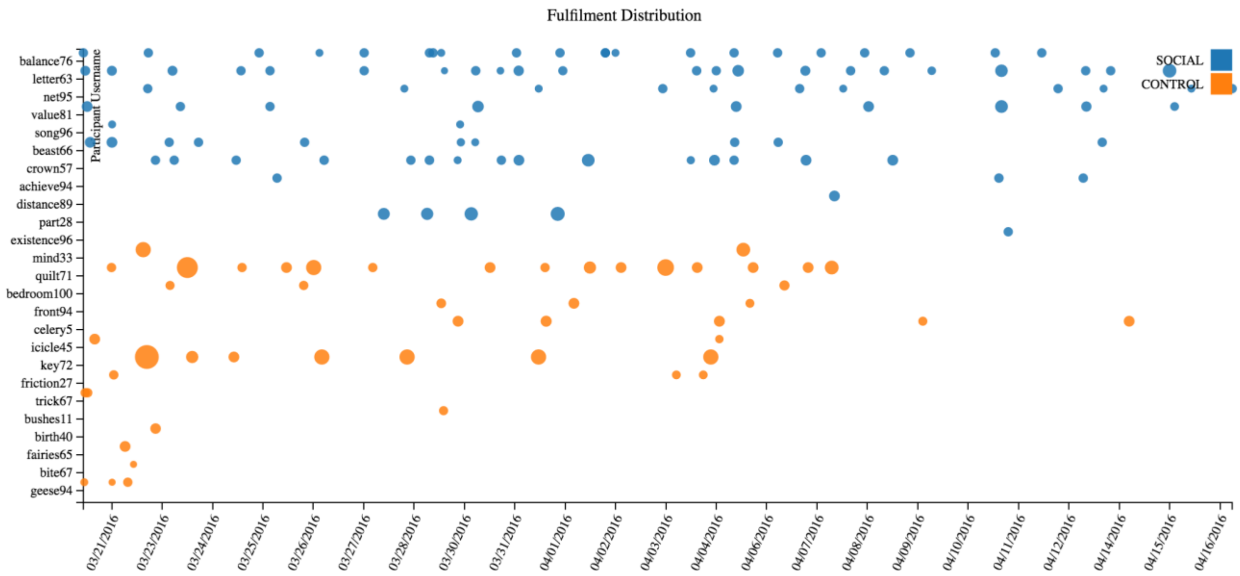


Figure 20: The distribution of fulfilments over the study period. The point radius is proportional to the duration of the fulfilment, where larger points indicate a longer fulfilment.

The fulfilment times for both cohorts were analysed and it was determined that participants in the SOCIAL cohort typically meditated for around 10 minutes, with a standard deviation of 4.5 minutes. The median and mode were both exactly 10 minutes, indicating a strong central tendency around this value, and a relatively small range of 23.5 minutes was found. The mean fulfilment time for the CONTROL cohort was considerably higher, at almost 17 minutes; the median and mode were also higher and further apart, the standard deviation of 11.25 minutes and the range of 58 minutes indicate a large variation between fulfilment times. This data can be seen in Table 6.

Cohort	Mean	Median	Mode	Range	Standard Deviation
SOCIAL	10.9	10	10	23.5	4.5
CONTROL	16.78	24	15	58	11.25

Table 6: Descriptive statistics for the fulfilment duration in both cohorts. The units are minutes.

The total fulfilment times for participants in both cohorts were also analysed. Participants in the SOCIAL cohort meditated for an average of 98 minutes whilst participants in the CONTROL cohort meditated, on average, for a considerably lower 58 minutes. The median for the SOCIAL cohort is also much higher and much closer to the average, indicating that the total activity of participants in the SOCIAL cohort varied less than those in the CONTROL cohort. The median value for the SOCIAL cohort is 24 minutes, less than half the average. This indicates that the total fulfilment times for participants in the CONTROL cohort varied by a large amount and that there were some participants far above the average and some far below.

Cohort	Mean	Median	Mode	Range	Standard Deviation
SOCIAL	98	88	10	239	81
CONTROL	58	24	-	288	85

Table 7: Descriptive statistics for total participant fulfilments in both cohorts. The units are minutes.

The total fulfilment times for participants in both cohorts were visualised and can be seen in Figure 21. The figure shows that the majority of participants in the SOCIAL cohort are adjacent to one another, surrounding the mean time of 98 minutes. The CONTROL cohort also features a prominent group around the mean of 58 minutes but with outliers at the highest and lowest end of the data. This agrees with the statement made above and shows that the total fulfilment times for participants in the CONTROL cohort varied considerably more than SOCIAL cohort.

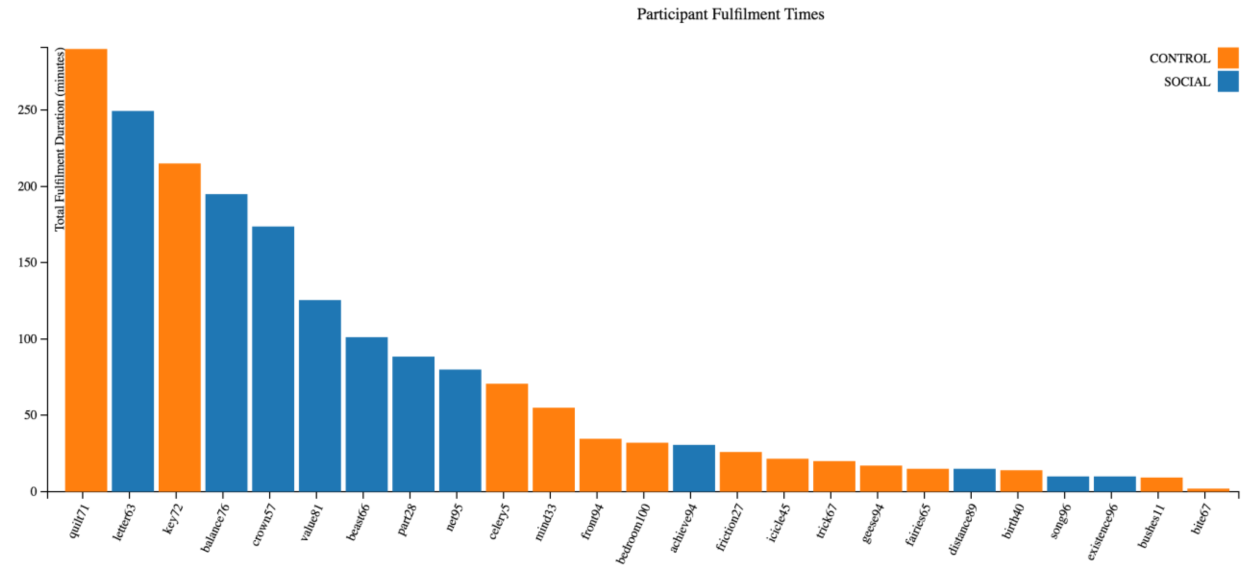


Figure 21: Total fulfilment times of participants in both cohorts.

Both parametric and nonparametric analysis was performed using the analysis platform. The tests were performed on all mediation fulfilments logged by participants, from both the SOCIAL and CONTROL cohorts. The analysis platform used, which is discussed in the Software section, contains the functionality to perform several nonparametric tests, a parametric test and to produce the descriptive statistics described above.

The fulfilment times were first analysed using one-factor ANOVA. The analysis of variance assumes normally-distributed data and tries to partition samples into groups which are compared to the distribution of the original data; this is discussed in more detail in the Analysis Platform section**.** A one-factor ANOVA was performed with 1 degree of freedom within-samples and 23 degrees of freedom between-samples giving an F-value of 1.363. A p-value was computed as 0.7404, indicating strong insignificance. The Brown-Forsythe test was also performed as it is more robust than ANOVA when non-Gaussian samples are used.

The Brown-Forsythe test resulted in an F-value of 0.333. A p-value was computed as 0.287 which also indicates insignificance, however, it is considerably less than that computed during ANOVA.

To perform nonparametric analysis, the Mann-Whitney U Test was used. This test does not assume normally-distributed data and, thus, overcomes some of the limitations of ANOVA when dealing with small sample sizes. The Mann-Whitney U Test resulted in a U-value of 53.5 for sample sizes of 11 and 14. The U-value was compared to a table of critical values (Critical Values of the Mann-Whitney U., 2016) and found to be insignificant as U < 46 for p = 0.05. This indicates that the probability of the results arising by chance is greater than 5%.

To find a corresponding p-value for the U-value a Monte Carlo simulation was sampled, a method found in the literature (Feltovich, Critical Values for the Robust Rank-Order Test, 2005). The simulation (lukem512/critical.js, 2016) was created by generating 1,000,000 samples of the correct sizes from a uniform distribution between 0 and 1 and ranking these using the Mann-Whitney U Test. The position of the sample with the same U-value was taken as the p-value. For a U-value of 54, a figure slightly higher than the reported U-value, which was not present, a p-value of 0.2019 was found. This indicates that the probability of the study results arising by chance is around than 20%.

The null hypothesis, that the differences between the fulfilment times of participants from the SOCIAL and CONTROL cohorts is likely to be a chance result, can therefore be tentatively rejected.

Qualitative Results

All participants from the SOCIAL cohort reported having meditated more during the study that previously. One participants from the CONTROL cohort reported having meditated less during the study, writing that it was “stress” that had prevented her from meditating and that “[using] the online app made me feel even more stressed because it was a reminder that I hadn’t been able to do it”.

A strong finding gathered from the questionnaires is that all participants, from both cohorts, reported an increase in motivation when scheduling meditation practices, as opposed to when they had not scheduled a practice. The motivation of participants to meditate, with and without scheduling a practice, has been presented in Figure 22; the motivation has been gauged using a Likert scale allowing it to be quantified. This finding was mirrored during interviews and one participant [SOCIAL, heavy meditator] commented “I enjoyed having access to the calendar to see what commitments were coming and it served as an excellent prompt and reminder to keep up with the plan”; another [SOCIAL, light meditator] saying “[I] enjoyed attending scheduled practise [sic]”; and yet another [SOCIAL, heavy meditator] saying “I like[d] being able to schedule meditations”.

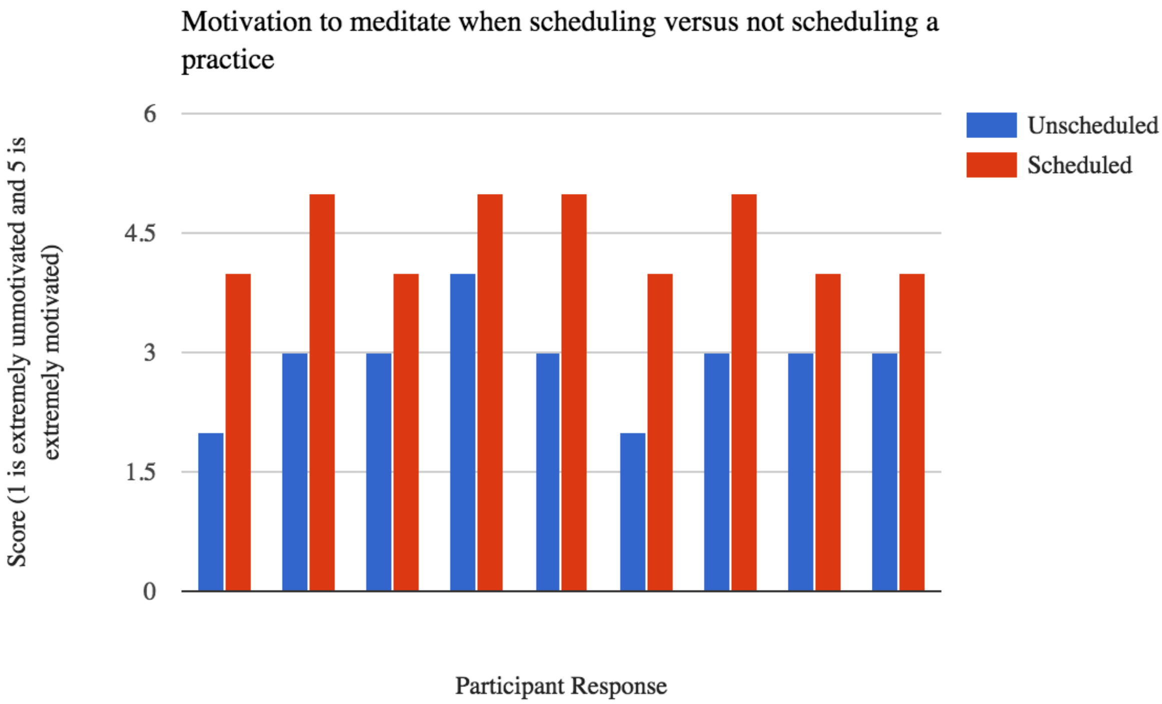


Figure 22: The reported motivation of participants to meditate when scheduling a practice as opposed to when they had not scheduled a practice.

Another strong finding is that participants in the SOCIAL cohort felt more strongly motivated to meditate when other participants were involved, either when they saw the activity of others on their feed (see Figure 23) or when other participants had pledged to join a schedule (see Figure 24). One participant [SOCIAL, average meditator] wrote “I found the interaction with other participants really helpful. It did make me want to meditate at times when others would be meditating, and although this was often not possible because of timings, seeing other participants schedules and activity on the feed definitely made it easier for me to motivate myself to meditate.” Another participant [CONTROL, heavy meditator] wrote that “[it] would have been nice to join in with others in some way.”

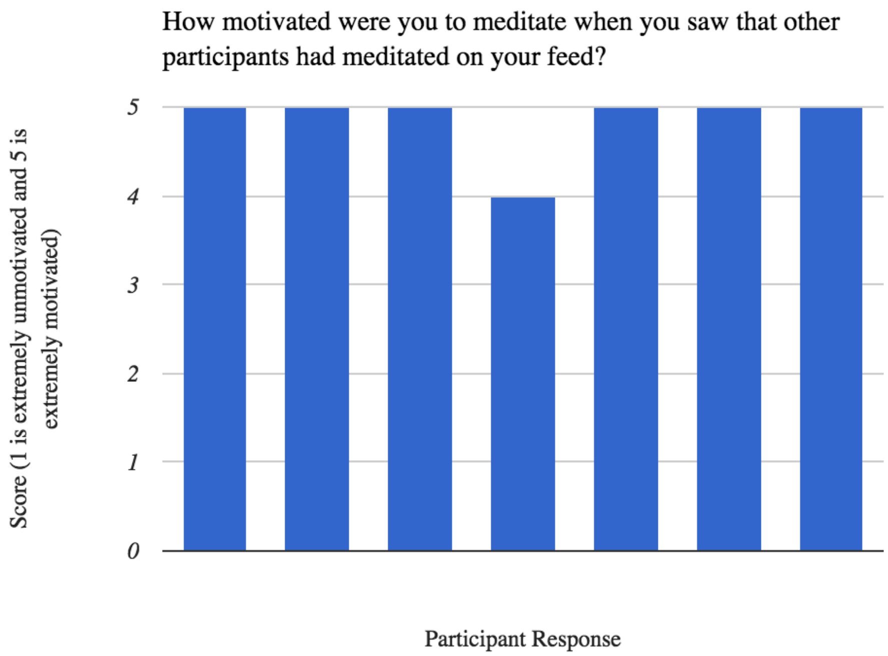


Figure 23: The reported motivation when seeing participant activity on the feed. This is the result of question 15 from the full questionnaire, as reported by participants in the SOCIAL cohort.

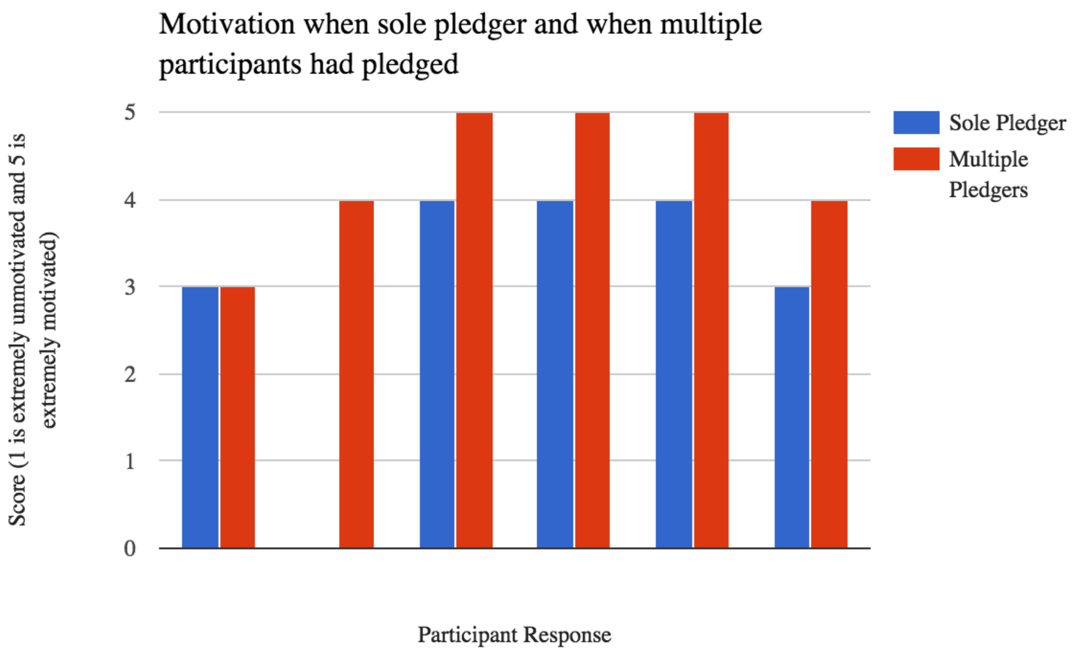


Figure 24: The reported motivation when a participant had scheduled a meditation (sole pledger) and when other participants had pledged to meditate on the same schedule (multiple pledgers). This is the result of questions 11 and 12 from the full questionnaire, as reported by participants in the SOCIAL cohort.

A result that runs contrary to this is that one participant [SOCIAL, average meditator] reported that they were unmotivated to pledge to join schedules created by other participants. However, the participant commented: “considering the impact that seeing other people’s sessions had, if I’d been more involved with the scheduling stuff I could see it being a big motivator. Unfortunately, due to poor observation on my part, by the time I realised I could create my own/join others’ schedules I already had an internal 'schedule' I was using.” The same participant also wrote “I only actually realised I could create schedules a couple of weeks ago” and reported being extremely motivated by seeing the activity of other participants on their feed.

Summary of Findings

- Participants in the SOCIAL cohort spent considerably more time meditating, on average, than participants in the CONTROL cohort
- Participants in the SOCIAL cohort logged more meditations than participants in the CONTROL cohort
- Participants in the SOCIAL cohort spent a similar time meditating in total than participants in the CONTROL cohort
- Participants in the SOCIAL cohort logged meditations throughout the entire study period whereas all but one participants in the CONTROL cohort had stopped by the end of the third week
- The duration of meditations logged by participants in the SOCIAL cohort varied considerably less than those logged by participants in the CONTROL cohort
- The nonparametric and parametric analysis were insignificant for small p-values, however, the nonparametric measure, which does not assume a normal distribution, was found to be significant for a p-value of 0.2.
- All respondents to the qualitative questionnaires claimed that they were more motivated to meditate when scheduling a practice
- Participants in the SOCIAL cohort felt more motivated to meditate when other participants were involved, either passively or actively

Analysis

Upon performing statistical tests on the fulfilments of participants the lowest reported p-value was 0.2, corresponding to a 20% probability of the data resulting from chance. The p-value reported when performing ANOVA was 0.7404 and indicates very little correlation between participants in each cohort. When performing Brown-Forsythe, a variant of ANOVA that is more robust to non-Gaussian data, the report p-value was reduced to 0.287. When performing the nonparametric method, the Mann-Whitney U Test, the p-value was further reduced to 0.2. These findings suggest that the data is unlikely to be normally-distribute, due to the observed improvement in p-values resulting from the Brown-Forsythe Test over ANOVA, and the further improvement resulting from the Mann-Whitney U Test over both parametric methods.

Due to the large p-values resulting from the statistical tests strong conclusions cannot be made; however, by using the qualitative feedback and inferring trends from the data, some initial answers to the questions surrounding the research suggested. It is suggested that these results be investigated in a follow-up study to confirm them as it is likely that the statistical insignificance is a result of the limited data set, in part due to the high drop-out rate of participants.

Initial analysis of the data, inferred by looking at the descriptive statistics such as the group means, suggests that the first hypothesis is true: the SOCIAL cohort meditated for a mean of 98 minutes whilst the CONTROL cohort meditated for a mean of 58 minutes. The variance of both cohorts is considerable, however, with the SOCIAL group varying by a standard deviation of 81 minutes and the CONTROL group varying by 85 minutes, an amount greater than the mean. As such, further analysis is needed before any conclusions can be drawn.

By observing the central tendencies of the total fulfilment times of participants in the two cohorts it seems that those in the SOCIAL cohort were less varied and closer to the mean: the median was close, only 10 minutes less than the average, and the range, whilst considerable, was less than that of the CONTROL cohort by over 20%. The data in Figure 21 seems to agree with this analysis as the majority of participants in the SOCIAL cohort are distributed around the mean and, whilst there are outliers, these are less polarised than those present in the CONTROL cohort.

By examining the data on the individual fulfilment times of participants in the two cohorts it also seems apparent that the fulfilments made by participants in the SOCIAL cohort were less varied, typically lasting 10 minutes and with a small range of around 20 minutes between the shortest and longest. The fulfilments made by participants in the CONTROL cohort, on the other hand, appear to be much more varied and the range and standard deviations are considerably larger.

A possible explanation for the reduced variance in the SOCIAL cohort is that participants new to meditation adopted the durations of other participants, or that participants creating initial schedules and logging fulfilments set the trend for those doing so later. This could be described as the emergence of social norms in the online platform, with the initial participants 'seeding' the behaviour that was adopted by others; this agrees with the findings of Postmes et al (Postmes, 2000)**.** Participants in the CONTROL cohort did not have other participants to base their practice schedules on and, as such, may have adopted one that suits them, either through trial-and-error, leading to more variation within each participant's times, or arbitrarily, leading to more variation between participants.

The participants in the SOCIAL cohort seem to have meditated more than those in the CONTROL cohort, as the average time fulfilled, and the average number of fulfilments, is considerably higher. The variation in the total time fulfilled is also smaller, albeit by a small amount. The variation in the individual fulfilment times is much smaller however, as discussed above, and seems to suggest that the second hypothesis is also true.

The distribution of participant fulfilments shows that, by the third week, all but one of the participants in the CONTROL cohort had stopped logging meditations. This is in stark contrast to the SOCIAL cohort in which 8 participants logged meditations in this period. The attrition rate for CONTROL participants is, as such, considerably higher. It is hypothesised that this is due to a lack of social acknowledgement, or reward, when completing a meditation. In the literature this is discussed by Gollwitzer and Mahler where they found that completing a task socially was more effective than completing one in isolation (Gollwitzer,

1986) (Mahler, 1933); it is possible that the social 'completion' of a meditation provides enough of a reward to overcome the additional difficulty of recording the practice using the platform.

Additionally, the lack of reward may have discouraged participants from continuing to use the platform, even if they continued to meditate, due to the additional work for no perceived benefit. One participant [CONTROL, heavy meditator] mentioned, during the qualitative feedback, that "having to fill in dates etc. to schedule and then to log [was] repetitive" whilst another [CONTROL, light meditator] said "I did forget to record my meditations sometimes - more towards the end of the study" and yet another saying "sometimes it seemed like hard work". Interestingly, one participant [SOCIAL, average meditator] stated "I suspect that over time and if not observed or the sense of being observed diminished, then I would feel less strongly about completing my commitments"; whilst the comment was negative it is also indicative that, if the level of social interaction on the platform increased, it would continue to be a motivation.

There were 4 participants in the CONTROL cohort that logged a single meditation, 3 of whom did so in the first 2 days and the final participant at the beginning of the second week. This could be explained by a lack of motivation to continue meditating once the initial 'scientific interest' had been satisfied, as suggested by Rotman et al (Rotman), as there was no way for their efforts to be acknowledged. The participants in the SOCIAL cohort typically meditated much more than once, as only 2 participants logged a single meditation, and could be explained conversely: once the initial interest had worn off this motivation could be replaced by acknowledgement as other participants in the cohort witnessing their activity. The 2 participants in the SOCIAL cohort that only meditated once did so towards the end of the study; this may be due to them logging onto the platform and being prompted to participate when viewing the activity of other participants, a cue not present to those in the CONTROL cohort.

The ability to make schedules was also well-received and, surprisingly, provided a lot of motivation for participants. This was reported in both the questionnaires and during participant interviews. By looking at Figure 20 we can see that, quantitatively, all participants that responded to the questionnaire agreed that they were more motivated if they had scheduled a practice. The increase in motivation when scheduling a practice in advance is not obviously linked to social norms, as all participants reported an increase, however the comments specifically mentioning the schedules were all made by participants in the SOCIAL cohort, either during interview or in a questionnaire question asking for "positive experiences".

These additional suggestions, which do not relate to the hypotheses specified prior to the study, may be useful in the design of digital platforms wishing to leverage social norms as a means of increasing participant motivation or interaction. This is discussed in more detail in the Design Recommendations section.

When considering which of these effects may generalise to an arbitrary task, beyond meditation, it seems that the adoption of a social norm is likely. The adoption of a social norm by an online community has been discussed in the literature (Postmes, 2000) and it appears to have occurred during this study, as described in the analysis above. The social norm, in this case, led to the adoption of a ‘standard’ practice time that was used by many participants; this is likely to be an effect of utilising the platform for arbitrary tasks and may be manipulated by providing initial schedules or participant data. It is also possible, however, that the norm may result in a competitive atmosphere in which the normal behaviour becomes outperforming the other participants - this is more likely to occur if the chosen task was competitive in nature, unlike meditation.

It also seems probable that the ability to schedule tasks in advance will increase motivation, regardless of the task chosen. This is due to the overwhelming response that participants gave when questioned about their motivation to meditate when they had and had not scheduled it, to which every participant claimed that it had strongly increased.

Finally, it seems likely that participants in a social environment will outperform those in an isolated environment for any task. Meditation is a solitary task and one generally not seen as benefiting from extrinsic motivation; as the results indicated that participants in the SOCIAL cohort meditated for more on average, and reported that social interaction was a strong motivator, it is probably that this effect will generalise for tasks more social in nature.

Summary of Analysis

- The data gathered by the study does not appear to be normally distributed, possibly due to a small sample size resulting from the drop-out rate
- Participants in the SOCIAL cohort seem to have meditated more and for less varied times than participants in the CONTROL cohort
- A social norm may have been adopted by participants in the SOCIAL cohort to schedule and log meditation practices of 10 minutes
- A higher attrition rate in the CONTROL cohort may be explained by a lack of social reward for using the platform
- Participants in the CONTROL cohort that logged a single or a few meditations in the initial study period may have failed to participate further due to a lack of additional motivation once the ‘scientific interest’ had been satisfied
- The creation of schedules dramatically increased the motivation of participants from both cohorts and seemed to particularly affect the participants in the SOCIAL cohort

Design Recommendations

As a result of the analysis several design recommendations have been made. These recommendations are appropriate for digital platforms wishing to encourage participant involvement by performing a task; they may, however, generalise to additional applications.

The first recommendation is that an activity feed is present, or an alternative manner of quickly allowing participants to share and become aware of their contributions and the contributions of others. This should be visible without participants having to ‘look for it’, maximising the social exposure. In the study platform the social feed replaces the home page for signed-in participants to achieve this. By using an activity, feed participants will be aware of the activity of others and will be aware that their own activity, including their pledges, is visible. The increase in motivation when observing the activity of other participants is considerable, as found in the questionnaire results (see Figure 23).

Another recommendation is to include a scheduling function as every questionnaire respondent reported an increase in motivation when scheduling a meditation as opposed to when they had not scheduled one. When the participant schedules a task, the participant is pledging to complete it; as such this integrates easily with a design incorporating social pledges. A suggestion made by several participants was that their schedules could be synchronised with their calendar in some way so that they could receive notifications and access their schedules on their mobile phones; including this functionality is likely to further increase participation as schedules will not be forgotten and their pledges reinforced when viewed subsequently.

A third recommendation is that displaying participant contributions socially is likely to decrease the attrition rate of participants and should be used even when the need to maximise individual participant motivation is not a concern. The analysis of the performed study suggests that a reason for the high attrition rate in the CONTROL cohort, in which participants were not isolated, could have resulted from a lack of acknowledgement; this can be easily avoided by using an activity feed or just displaying the contributions of a participant on a profile page.

Follow-up Studies

In the Technical Background section several open research questions were posed, the initial answers to 2 of which have been tentatively given in the results of the study performed during this dissertation. Unfortunately, due to the high rate of participant drop-out, the quantitative data gathered was not significant and the answers to these questions have been inferred from qualitative data and examining the general trends observed in the data. To confirm these initial answers the study should be repeated with larger sample sizes and the results re-analysed. This can be easily performed by using the study platform and following the deployment documentation, found on the GitHub repository for the project (lukem512/zen - GitHub, 2016), and by following the experimental method discussed above.

A second follow-up study of interest is to repeat the procedure with a substitute task and to observe whether the trends vary. This would be of particular interest for a competitive task, such as a sport or a game, as the resulting social norm may involve competing with the other participants rather than adopting a similar practice pattern, as was the case when meditation was used. A repeat study with a substitute task can be very simply accomplished by modifying the dictionary section configuration file; an example, given in the Study Platform section, can be seen in Code Listing 1.

Another follow-up study that explores an open research question is to hypothesise about the similarity between a social pledge and a social completion event, as proposed by Gollwitzer and Mahler (Gollwitzer, 1986) (Mahler, 1933). If the effect of pledging socially is similar to completing a task socially then it may discourage participants from actually completing the pledged activity. This hypothesis could be studied by modifying the platform to display only participant fulfilments rather than schedules and pledges, ensuring that the only social interaction between participants is when a task has been performed rather than the intention simply being stated.

Leading on from the question of similarity between a social pledge and social completion is the question of whether the demotivation of a social completion event can be overridden by including negative feedback in the activity feed: for example, by including items that inform other participants that a participant has failed to attend a schedule or did not complete their pledge. This may have implications in the design of a platform that utilises social pledges for motivation as, if the former question is true, a form of negative feedback would have to be included to leverage the positive, motivating effects of social pledges.

A final follow-up study is one that explores the manipulation of social norms by 'seeding' the initial activity on the platform. In the Analysis section it is proposed that the social norm of meditating for 10 minutes was adopted by later participants as a result of earlier participant behaviour; this can be explored by designing a study in which initial behaviours are established by the researchers and are modified for different cohorts. This could be achieved by including one or more researchers within each cohort that schedule and fulfil activities for a set or similar amount of time over an initial period and, once the study has been concluded, by observing the average fulfilment durations and spread and comparing these to the seeded activity.

Conclusion

This dissertation examined the emerging role of gamification in the design of digital products, platforms and communities, and a literature review of relevant academic papers on social norms and social pledges was performed. During this review several open research questions in these fields were identified including whether the use of social pledges can provide additional motivation for members of a community or whether they may take on a similar role to a social completion and demotivate, and whether social pledges may encourage the adoption of novel social norms and behaviours.

To allow these research questions to be explored a generic study platform was implemented, allowing participants to pledge and record their completion of an arbitrary task. The platform was designed to allow a range of situations to studied including the separation of participants into social or isolated cohorts. The platform also allows for the appearance and content to be customised, without requiring technical knowledge of the platform, via a configuration file.

An analysis platform was also created; this integrates with the study platform and exposes the ability to perform multiple statistical methods including the analysis of variance and the Mann-Whitney U Test. When these tests are performed the results are displayed alongside a basic interpretation and instructions about the meaning of the output and how to include any findings in a report.

The software created has been designed to be user-friendly, generic and to provide a complete solution for conducting studies into social pledges. A researcher can use the platform to generate anonymous participant credentials, assign participants to cohorts, gather quantitative data and then analyse the data all via an intuitive, online interface. The software has also been designed to be extensible, providing scope for its use beyond the currently conceived follow-up studies.

A study was conducted into two of the open research questions: whether the motivation of participants was increased by pledging socially and whether a social norm would be adopted, leading to a trend in participant behaviour. The study showed promising results and seemed to indicate that the initial hypotheses were correct, although, due to a small sample size resulting from a high drop-out rate, additional study is needed to confirm these findings. Despite this, initial analysis, performed in part using qualitative data, suggests that participants pledging socially did experience an increase in motivation and that the fulfilments of participants aware of the behaviour of others typically varied less within- and between-participants, indicating the emergence of a social norm. This begins to answer some open research questions in social pledging, a valuable field for research within HCI.

Additional findings were also made, unrelated to the questions posed at the beginning of this dissertation, that may have implications in the design of products utilising social pledges and social norms. These findings, including the strongly motivating effect of scheduling an activity in advance, have been discussed and recommendations for designs employing them have been given.

Several follow-up studies have been designed to confirm the findings made during the study and to build upon them. These follow-up studies can all be performed using the study platform with a minimal modification. The studies aim to confirm the answers to the questions that have been tentatively given as a result of the study performed and, additionally, aim to answer the outstanding open research questions.

The study was performed using an instantiated version of the study platform and analysed using the study platform; in doing so the function of the software has been verified. Several small improvements were made to the platform as a result of feedback by study participants and, upon examining feedback given during participant interviews and from questionnaires, the experiences were overwhelmingly positive. There is scope for additional improvement to be made to the platform and this has been discussed in the Future Work section; the improvements will enable the software to be used for an even broader range of studies.

Finally, in the duration of this dissertation 5 open-source components for performing statistical analysis, including 2 JavaScript modules that provided functionality as-of-yet unavailable to the language, were created. Additionally, during the creation of this document, an improvement was made to an open-source LaTeX styling tool (Works with Python 2.x and 3.x - LaTeX-Bibitem-Styler/latex-bibitemstyler, 2016).

Bibliography

1. *(X)HTML Validation results for zen.lukemitchell.co.uk*. (2016). Retrieved from W3C: https://html5.validator.nu/?doc=https%3A%2F%2Fzen.lukemitchell.co.uk%2F
2. *Analysis of Variance (ANOVA)*. (2016). Retrieved from Plymouth University: http://www.cimt.plymouth.ac.uk/projects/mepres/alevel/fstats_ch7.pdf
3. *Anatomy of a "goto fail" – Apple's SSL bug explained, plus an unofficial patch for OS X!* (2014, February 24). Retrieved from Sophos: https://nakedsecurity.sophos.com/2014/02/24/anatomy-of-a-goto-fail-apples-ssl-bug-explained-plus-an-unofficial-patch/
4. *anova*. (2016, April 18). Retrieved from NPM: https://www.npmjs.com/package/anova
5. Behr, M. G. (2013, May 7). *Under the surface of the NY Times mobile redesign*. Retrieved from Open News: https://source.opennews.org/en-US/articles/nyt-redesigns-mobile/
6. *brown-forsythe-test*. (2016, April 18). Retrieved from NPM: https://www.npmjs.com/package/brown-forsythe-test
7. Bunch Ball. (2016). *Bunch Ball*. Retrieved April 20, 2016, from Bunch Ball: http://bunchball.com
8. *Cerulean Theme*. (2016). Retrieved from BootSwatch: https://bootswatch.com/cerulean
9. Chen, Y. H. (2010). Social comparisons and contributions to online communities: A field experiment on MovieLens. *The American Economic Review, 100*(4), pp. 1358-1398.
10. Cialdini, R. R. (1990). A focus theory of normative conduct: Recycling the concept of norms to reduce littering in public places. *Journal of Personality and Social Psychology, 58*(6), pp. 1015-1026.
11. *Cloud Application Platform | Heroku*. (2016). Retrieved from Heroku: https://www.heroku.com/
12. *Critical Values of the Mann-Whitney U.* (2016). Retrieved from UMass Boston: http://ocw.umb.edu/psychology/psych-270/other-materials/RelativeResourceManager.pdf
13. de-Marcos, L. D.-d.-N. (2014). An Empirical Study Comparing Gamification and Social Networking on e-Learning. *Computers & Education*.
14. Deterding. (2011, May 7-12). Gamification: Using Game Design Elements in Non-Gaming Contexts. *CHI ACM 978-1-4503-0268.*
15. *Differences Between Fast Hashes and Slow Hashes*. (2012, July 7). Retrieved from OpenWall: http://openwall.info/wiki/john/essays/fast-and-slow-hashes
16. Duffy, J. a. (2010). Does competition affect giving? *Economic Behaviour and Organisation, 74*(1-2), pp. 82-103.
17. *Duolingo: Learn Spanish, French and other languages for free*. (2016, April 28). Retrieved from Duolingo: https://www.duolingo.com/
18. Enki. (2016). *GitHub - enkidevs*. Retrieved from GitHub: https://github.com/enkidevs
19. Feltovich, N. (2003). Nonparametric differences in Wilcoxon-Mann-Whitney and Robust Rank-Order Tests. *Experimental Economics, 6*, pp. 273-297.

20. Feltovich, N. (2005). Critical Values for the Robust Rank-Order Test. *Communications in Statistics - Simulation and Computation, 34*(3), pp. 525-547.
21. Ferenstein, G. (n.d.). How social gaming is improving education.
22. Foundation, N. (2016, March 22). *How Uber Uses Node.js To Scale Their Business*. Retrieved from Medium: https://medium.com/@nodejs/how-uber-uses-node-js-to-scale-their-business-cd4d8f8e4187#.u31epiwgs
23. *Freelancer Theme*. (2016). Retrieved from Start Bootstrap: http://startbootstrap.com/template-overviews/freelancer/
24. FullCalendar. (2016). *FullCalendar - JavaScript Event Calendar (jQuery)*. Retrieved from FullCalendar: http://fullcalendar.io/
25. *GetBadges - gamification platform for developers*. (2016). Retrieved from GetBadges: https://getbadges.io/
26. *getX functions for Year, Month and Day - Pull Request #591 - Serhioromano/bootstrap-calendar*. (2016, February 12). Retrieved from GitHub: https://github.com/Serhioromano/bootstrap-calendar/pull/591
27. Glover, I. (2013). Play As You Learn: Gamification as a Technique for Motivating Learners. *Proceedings of World Conference on Educational Multimedia* (pp. 1999-2008). Chesapeake, VA: Hypermedia and Telecommunications.
28. Gollwitzer, P. M. (1986). *Striving for specific identities: The social reality of self-symbolizing.* New York: Springer.
29. Google, I. (2016, April 30). *Web Search Interest "gamification" 2004-Present*. Retrieved from Google Trends: https://www.google.co.uk/trends/explore#q=gamification
30. Harland, P. S. (1999). Explaining proenvironmental intention and behavior by personal norms and the Theory of Planned Behavior. *Journal of Applied Social Psychology, 29*(12), pp. 2505-2528.
31. Herlihy, P. (2013, October 21). *How many people are missing out on JavaScript enhancement?* Retrieved from Government Digital Services: https://gds.blog.gov.uk/2013/10/21/how-many-people-are-missing-out-on-javascript-enhancement/
32. Howell, D. C. (2002). *Statistical Methods for Psychology.* Duxbury/Thomson Learning.
33. IANA. (2016, March 01). *Hypertext Transfer Protocol (HTTP) Status Code Registry.* Retrieved from Internet Assigned Numbers Authority: https://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml
34. *Internet Access - Households and Individuals*. (2015). Retrieved from Office for National Statistics: http://www.ons.gov.uk/peoplepopulationandcommunity/householdcharacteristics/homeinternetandsocialmediausage/bulletins/internetaccesshouseholdsandindividuals/2015-08-06
35. *JavaScript Standard Style*. (2016). Retrieved from StandardJS: http://standardjs.com
36. *JSON Web Tokens*. (2016). Retrieved from JSON Web Tokens: https://jwt.io/introduction/
37. Kirkpatrick, D. (2013, March 21). *Mongodb - Security Weaknesses in a typical NoSQL database*. Retrieved from TrustWave: https://www.trustwave.com/Resources/SpiderLabs-Blog/Mongodb---Security-Weaknesses-in-a-typical-NoSQL-database

38. *levene-test*. (2016, April 18). Retrieved from NPM: https://www.npmjs.com/package/levene-test
39. Li, W. G. (2012). GamiCAD: A Gamified Tutorial System For First Time AutoCAD Users. *UIST*.
40. *Luke (@LukeCrypto)*. (2016). Retrieved from Twitter: https://twitter.com/lukecrypto
41. *lukem512/critical.js*. (2016, May 1). Retrieved from GitHub Gist: https://gist.github.com/lukem512/f8dca8b7b499cf1e667752e4e3b86db6
42. *lukem512/zen - GitHub*. (2016). Retrieved from GitHub: https://github.com/lukem512/zen
43. *lukem512/zen alternative branch*. (2016). Retrieved from https://github.com/lukem512/zen/tree/alternative
44. Mahler, W. (1933). Ersatzhandlungen verschiedeneir Realitätsgrades [Substitute actions at various levels of reality]. *l Psychologische Forschung, 18*, pp. 27-89.
45. *mann-whitney-utest*. (2016, April 18). Retrieved from NPM: https://www.npmjs.com/package/mann-whitney-utest
46. Marczewski, A. (2012, April). *Gamification: A Simple Introduction.*
47. Massung, E. &. (2014). Normification: using crowdsourced technology to affect third-party change. *ACM CHI EA*, (pp. 1449-54).
48. Matthews, G. (2013, January 3). *Goals Research Summary*. Retrieved from Dominican University of California: http://www.dominican.edu/academics/ahss/undergraduate-programs/psych/faculty/assets-gail-matthews/researchsummary2.pdf
49. *MEAN.io - The Friendly & Fun Javascript Fullstack for your next web application*. (2016). Retrieved from MEAN.io: https://mean.io
50. *Module Counts*. (2016). Retrieved from Module Counts: http://www.modulecounts.com/
51. Moment.js. (2016). *Moment.js*. Retrieved from Moment.js: http://momentjs.com/
52. *MongoDB Hosting - Database-as-a-Service by mLab*. (2016). Retrieved from mLab: https://mlab.com/
53. Postmes, T. S. (2000). The formation of group norms in computer-mediated communication. *Human Communication Research, 26*, pp. 341-371.
54. Preist, C. M. (2014). Normification as a Means of Engaging Digital Volunteers. *Volunteering and Doing Good* (pp. 1222-1233). CSCW.
55. *Promise or Pay*. (2016). Retrieved from Promise or Pay: http://promiseorpay.com/
56. *RFC 7519 - JSON Web Token (JWT)*. (2015, May). Retrieved from Internet Engineering Task Force: https://tools.ietf.org/html/rfc7519
57. Rimal, R. a. (2005). How behaviors are influenced by perceived norms: A test of the theory of normative social behavior. *Communication Research*, *32*(3), pp. 389-414.
58. Rotman, D. P. (n.d.). Dynamic changes in motivation in collaborative citizen-science projects. *ACM. 2012*, pp. 217-226. CSCW.
59. *SAP Software Solutions*. (2016). Retrieved from SAP Software Solutions: http://go.sap.com/index.html
60. Schultz, P. N. (2007). The constructive, destructive, and reconstructive power of social norms. *Psychological Science, 18*(5), pp. 429-434.
61. Schultz, W. K. (2008). Using normative social influence to promote conservation among hotel guests. *Social Influence, 3*(1), pp. 4-23.
62. *Security Checklist*. (2016). Retrieved from MongoDB: https://docs.mongodb.org/manual/administration/security-checklist/

63. Simões, J. R. (2013). A social gamification framework for a K-6 learning platform. *Computers in Human Behavior, 29*(2), pp. 345-353.
64. *Soccer+*. (2016). Retrieved from Soccer+ (Alternative Study Platform Instance): https://serene-ravine-41117.herokuapp.com
65. *statistical-methods*. (2016, April 18). Retrieved from NPM: https://www.npmjs.com/package/statistical-methods
66. *Study Platform Deployment with Alternative Skin*. (2016). Retrieved from https://safe-reaches-25059.herokuapp.com/
67. *Support for older versions of Internet Explorer ends. Microsoft*. (2016, January 12). Retrieved from Microsoft: https://www.microsoft.com/en-gb/WindowsForBusiness/End-of-IE-support
68. Tally, S. (2012, 11 27). *Digital badges show students' skills along with degree*. Retrieved from Purdue University: http://www.purdue.edu/newsroom/releases/2012/Q3/digital-badges-show-students-skills-along-with-degree.html
69. *Twitter Bootstrap*. (2016). Retrieved from Twitter Bootstrap: http://getbootstrap.com/
70. Uchicha, M. (2012, September 15). *Why Would One Bother Marking Up Properly and Semantically?* Retrieved from Stack Exchange: https://programmers.stackexchange.com/questions/164988/why-would-one-bother-marking-up-properly-and-semantically.
71. Voyer-Perrault, G. (2016, February 28). *What are the biggest websites built with Node.js on the server side?* Retrieved from Quora: https://www.quora.com/What-are-the-biggest-websites-built-with-Node-js-on-the-server-side.
72. *Weight Watchers UK*. (2016, April 28). Retrieved April 28, 2016, from Weight Watchers: ttps://www.weightwatchers.com/uk/
73. *Works with Python 2.x and 3.x - LaTeX-Bibitem-Styler/latex-bibitemstyler*. (2016, January 21). Retrieved from GitHub: https://github.com/LaTeX-Bibitem-Styler/latex-bibitemstyler/commit/8c596582df4d59ccfd57265f142ee59d0cc4796c
74. *Zen (Staging)*. (2016). Retrieved from http://test.zen.lukemitchell.co.uk/

Appendix A: Initial UI Testing Checklist

Zen: Initial UI Testing

To access the website you should visit https://boiling-ravine-11892.herokuapp.com/.

Please attempt to perform the following tasks and make a note of any difficulties you experienced and inform me; if you feel that there is a more intuitive way to perform a task please mention this too. Thank you for helping!

1. Sign into the website using the following credentials:

Username:	**Password:**

2. Schedule a meditation session for one day next week. The session can be for as long as you wish, at any time or day. 3. Schedule a meditation session for another day next week. The session should be at 12:00 and it should last one hour.

4. Edit the meditation session from part 3. It should now start at 11:00 and it should last one hour and 30 minutes.

5. Find a meditation session that someone else has scheduled. View the schedule.

6. Pledge to join the meditation schedule from part 5.

7. View the profile of the participant that created the schedule from part 5.

8. Log a meditation session that you performed yesterday afternoon. The session should last for 15 minutes and can begin at any time between 12:00 and 18:00.

9. Edit the meditation session from part 9 to change to duration to 30 minutes.

10. Begin a meditation session and ‘meditate’ for 30 seconds.

11. Change your password to a phrase of your choice.

12. Find the contact details for the website and email me your feedback.

Appendix B: Questionnaire

Name:

Age:

Gender: Male / Female / Prefer not to say

Location:

Occupation:

1. Had you meditated prior to participating in the study?

2. If so, did you find that you meditated more or less during the study?

3. What was your main motivation for meditating during the study?

4. Did you have any positive experiences using the online application?

5. Did you have any negative experiences using the online application?

6. How motivated were you to meditate when you had not scheduled a practise?

Please select the score most appropriate, where 1 is extremely unmotivated and 5 is extremely motivated.

1 2 3 4 5 N/A

7. How motivated were you to meditate when you had scheduled a practise?

Please select the score most appropriate, where 1 is extremely unmotivated and 5 is extremely motivated.

1 2 3 4 5 N/A

8. Any other comments?

Thank you!

Addendum for participants in SOCIAL cohort

9. How motivated were you to create schedules during the study?

Please select the score most appropriate, where 1 is extremely unmotivated and 5 is extremely motivated.

1 2 3 4 5 N/A

10. How motivated were you to pledge to join schedules during the study?

Please select the score most appropriate, where 1 is extremely unmotivated and 5 is extremely motivated.

1 2 3 4 5 N/A

11. How motivated were you to meditate when you were the only participant pledged to a schedule?

Please select the score most appropriate, where 1 is extremely unmotivated and 5 is extremely motivated.

1 2 3 4 5 N/A

12. How motivated were you to meditate when there were other participants pledged to a schedule?

Please select the score most appropriate, where 1 is extremely unmotivated and 5 is extremely motivated.

1 2 3 4 5 N/A

13. How motivated were you to meditate when you were pledged to another participant's schedule?

Please select the score most appropriate, where 1 is extremely unmotivated and 5 is extremely motivated.

1 2 3 4 5 N/A

14. How motivated were you to meditate when there were other participants pledged to your schedule?

Please select the score most appropriate, where 1 is extremely unmotivated and 5 is extremely motivated.

1 2 3 4 5 N/A

15. How motivated were you to meditate when you saw that other participants had meditated on your feed?

Please select the score most appropriate, where 1 is extremely unmotivated and 5 is extremely motivated.

1 2 3 4 5 N/A

16. How motivated were you to pledge to meditate (join a schedule made by another participant) when you saw that other participants had pledged to meditate on your feed?

Please select the score most appropriate, where 1 is extremely unmotivated and 5 is extremely motivated.

1 2 3 4 5 N/A

17. Do you have any comments relating to the interactions with other participants on the application?

Thank you!