*A****bstract— <placeholder>***

**Sleepify: A system towards personalizing and optimising sleeping environments**

**Jeremy Chan, Tsz Ho Ho, Dominic Kwok, Ho Shun Lo, Nathalie Wong**

Department of Electrical and Electronic Engineering, Imperial College London, SW7 2AZ  
Email: {jc4913, thh13, cyk113, hsl113, nw813}@ic.ac.uk

# Introduction

It is

# Sleepify’s Promise

This project aims to provide a better sleeping experience overall from having the room temperature automatically adjust to body and room temperature information from two sensors. Sleepify also promises improved performance and improved machine learning classification accuracy based on prolonged usage of the app. Continued usage of Sleepify is especially important for our machine learning algorithms; thankfully the retention rate of health and fitness apps are the highest among others [1]. Lastly, Sleepify promises to deliver a slick and intuitive app, and web interface for the user to use and interact with, motivating the user to continue using Sleepify regularly; this is crucial to having a low app abandonment rate [2].

# Related Work

There are many sleep trackers on the market that use a variety of ways to track sleep quality. Most sleep trackers monitor the user’s different stage of sleeping, sleeping environment and provide sleep coaching advice. Majority of the trackers are found in the form of software application for iOS and Android. These applications use the accelerometer found in smartphones to track body movement throughout the sleep cycle. Using this data, “Sleep Cycle” wakes the user up during the lightest sleep phase, preventing the feeling of tiredness in the morning. In addition to the accelerometer, “Sleep as Android” [3] records audio through the microphone to detect snoring, speech, and ambient noise. This can be played back to the user the following morning, and can be a good indicator of sleep disturbances and stress [4]. Additionally, some applications also include the feature of playing soothing sound or music to make the user fall asleep peacefully.

Hardware sleep trackers such as “S+ By ResMed personal sleep solution” contain even more features, such as synchronizing the output music with the respiratory pattern of the user to provide a calming effect [5]. Another interesting feature by “Aura Smart Sleep system” includes a red light to induce the user into sleep [6]. “Sense” has a slow wake up light alarm to gradually wake the user up. Most of the aforementioned also have questionnaires for the user to record their daily behavior to help analyze their sleeping pattern.

However, some of the down sides of these applications include inaccuracy in telling whether the user is just lying in bed or actually sleeping. Some drain the battery of both the device or the phone quickly. Some of the applications lack a snooze alarm function.

Sleepify has taken into account the pros and cons of these existing sleep trackers in the market when prioritizing its aims. In addition to the generic functions such as sleep coaching advice and sleep environment monitoring, it has taken an active role to provide a novel edge to sleep tracking - adjusting the sleeping environment. Sleepify analyses the best sleeping temperature and connects to smart heating devices to adjust the optimum sleeping temperature automatically. Manually changing the start time of the sleep record would also be enabled to prevent the problem of false sleep detection.

# System Design and Implementation

## Overall High Level Design

There are many types of software distribution models. Traditionally, users purchase a piece of software either through a retailer or online, and then install it onto their computer. The user then holds the license to use this piece of software, indefinitely. A drawback of this traditional method is that the user normally has to pay an upfront cost, update availability is subject to the package the user bought, and data only exists locally on the user’s machine [7].

Nowadays, the Software-as-a-Service (SAAS) model is the model upon which most companies are building their products around [8]. The SAAS model gives the consumer the ability to use on demand software that is provided by developer via the web or an app. As the user normally pays a subscription fee instead of an upfront fee, the SAAS model guarantees that the user will always be using the most updated version of the software as there is no ‘local copy’ of the software to install [9]. Moreover, SAAS removes the burden of having to configure (and control of) infrastructure for the user. However, SAAS solutions often assume that customers will always like new changes as updates are rolled out to all users [10] – this is not applicable to Sleepify as the timeframe of the project means only developing a minimal viable product (MVP).

Following the SAAS model, Sleepify consists of a front-end and a back-end, each consisting of two parts. The front-end is what the user sees and uses, and consists of an iOS application and the web interface. Updates through the App Store and the website ensure the user will always be using the most updated version of Sleepify. Finally, this front-end connects to the sensors for data collection and temperature adjustment. The back-end consists of the servers, databases, APIs, and machine learning modules – these both provide, and accept information from the front-end applications. The user has no information or control on how the back-end is configured; they need not to.

## Sensors

## Backend (Server, Database, API)

The backend is responsible for interfacing with the front-end, in accepting and providing it with the information it needs. It consists of a server on which a database resides, and a Representational State Transfer (RESTful) API which allows the iOS app, web interface, and machine learning sections to communicate with the server and by extension, the database.

### The Server

As the database, the web interface, and the API all reside upon the server, a smart choice needed to be made regarding how the server would be implemented.

Our group had prior experience in setting up a server running a LAMP stack (Linux, Apache, MySQL, PHP) in hacking together a simple custom API and website, but this was judged to be inadequate for Sleepify as trying to hand code PHP without a web framework when creating any sort of advanced web app would take an extremely long time. Laravel, and Yii, both modern PHP frameworks, were initially shortlisted as using a modern web framework would shorten development times drastically. However, the verbose and sometimes confusing syntax of PHP mean getting-things-done is more important than code readability [11]. As Sleepify’s development may continue in the future, reusability and code readability meant the group decided not to go with a PHP framework. Further research in hindsight also showed PHP to suffer from more security issues compared to other web frameworks [12].

Emphasising code readability and rapid development meant the choice was narrowed down to two programming languages: Python, and Ruby. The most well-known web framework in Python is Django, and in Ruby, Ruby on Rails. Both offer extremely fast prototyping and development, extensive documentation, security measures against common attacks, and multiple libraries to assist development. The final decision was to use Django, the Python web framework, as the ease of use of Python (especially considering the group had extensive C++ and Python experience) and the ample documentation on Django meant decreasing the time needed to create the MVP.

Multiple Django libraries were leveraged to add extra functionality, the most notable being: django-rest-framework, a library which provides the skeleton of the RESTful API; django-rest-auth, which extends Django’s already excellent authentication system with the API; django-bootstrap3, a library which simplifies styling a website using Twitter bootstrap.

### The Database

### The API

## Machine Learning

## iOS Application

## Web Interface

# Evaluation Criteria and Setup

# Results

# Discussion

# Conclusion

In conclusion, this report highlighted the motivation behind in building a system that is capable to alter the users thermal sleep environment to achieve better sleep quality. We have identified that the thermal environment is a key factor in affecting sleep quality, this justifies our rational in controlling this factor in order to provide better sleep quality to users. We have discussed related works, however to the authors’ knowledge there is no work that has developed a complete system to alter sleeping environments. Finally, we have also presented some of our prelimiarly works.

# References

[1] ‘Enter the Matrix: App Retention and Engagement’, *Flurry Blog*. [Online]. Available: http://flurrymobile.tumblr.com/post/144245637325/appmatrix. [Accessed: 19-Mar-2017].

[2] ‘App Retention Improves - Apps Used Only Once Declines to 20%’. [Online]. Available: http://info.localytics.com/blog/app-retention-improves. [Accessed: 19-Mar-2017].

[3] U. Team, *Sleep as Android Unlock*. Urbandroid Team, 2016.

[4] M. M. Ohayon and C. M. Shapiro, ‘Sleep disturbances and psychiatric disorders associated with posttraumatic stress disorder in the general population’, *Compr. Psychiatry*, vol. 41, no. 6, pp. 469–478, Nov. 2000.

[5] ‘S+ by ResMed’. [Online]. Available: https://sleep.mysplus.com/. [Accessed: 01-Feb-2017].

[6] ‘Withings’. [Online]. Available: https://www.withings.com/uk/en/products/aura. [Accessed: 01-Feb-2017].

[7] ‘What is SaaS’, *Interoute*, 01-Mar-2017. [Online]. Available: http://www.interoute.com/what-saas. [Accessed: 19-Mar-2017].

[8] J. P. • M. 30 and 2016, ‘Top 25 Most Popular SaaS, Cloud Applications for Business’, *ChannelE2E*, 30-Mar-2016. [Online]. Available: https://www.channele2e.com/2016/03/30/top-25-most-popular-saas-and-cloud-applications/. [Accessed: 19-Mar-2017].

[9] A. Heredia, R. Colomo-Palacios, and A. de Amescua, ‘Software Business Models from a Distribution Perspective: A Systematic Mapping Study’, *Procedia Comput. Sci.*, vol. 64, pp. 395–402, Jan. 2015.

[10] ‘Weighing the Pros and Cons of a SaaS Solution’, *AT&T Networking Exchange Blog*, 28-Jul-2016. [Online]. Available: https://networkingexchangeblog.att.com/enterprise-business/weighing-pros-cons-saas-solution/. [Accessed: 19-Mar-2017].

[11] S. Das, ‘Which is Better, PHP or Python? A Developer’s Take’, *LinkedIn Pulse*, 11-Jun-2015. [Online]. Available: https://www.linkedin.com/pulse/which-better-php-python-developers-take-srikrishna-das. [Accessed: 19-Mar-2017].

[12] S. M. Srinivasan and R. S. Sangwan, ‘Web App Security: A Comparison and Categorization of Testing Frameworks’, *IEEE Softw.*, vol. 34, no. 1, pp. 99–102, Jan. 2017.