ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 3, pp: (269-279), Month: July - September 2014, Available at: www.researchpublish.com

FATIGUE RISK MANAGEMENT USING MOBILE APPLICATION

[1]Ms. A. Sivasankari, [2]N. Karthika

[1,2]Department of Computer Science, D.K.M College for Women, Vellore, TamilNadu, India

Abstract: **Safety and productivity in the work-place are intimately related to worker health. A workplace may have chemical, physical, biological, and/or psychosocial hazards that have the potential to impact physical and psychological well-being. If the development of the iPhone application Crew Alert. The application is intended for use by pilots as a tool for assessment, logging and reporting of fatigue, to increase safety in the air. It also keeps track of the user's schedule, and sleep pattern. The application is developed from being only a iPhone application to being a fully universal iOS application supporting all iOS devices and screen orientations. Focus of development lies on usability and feature set, using the Goal Directed Design approach in combination with the Scrum framework. As development progressed the focus was put towards lowering the learning curve and creating more substance for the users in everyday use.**

Keywords: **Crew Alert, iphone, ios, security.**

1. INTRODUCTION

Fatigue may be defined as a physiological state of reduced mental or physical performance capability resulting from sleep loss or extended wakefulness, circadian phase, or workload (mental and/or physical activity) that can impair a person’s ability to perform their duties efficiently and safely.

Fatigue has been identified as a major area of risk for many workers in our modern 24 hour society and a significant source of workplace accidents and incidents. Many jurisdictions have legislated that fatigue is a hazard that must be managed in the workplace. Fatigue impacts many aspects of a business’ operations, because of its effects on mood, communication, reaction times, alertness and decision making.

Employees are also affected by fatigue on a personal level, both at home and at work. This can lead to communication breakdowns and anti-social behavior at home, and dissatisfaction and absenteeism in the workplace. Job satisfaction and staff retention may then become an issue for employers.

Safety and productivity in the workplace are intimately related to worker health. A workplace may have chemical, physical, biological, and/or psychosocial hazards that have the potential to impact physical and psychological well-being. How these hazards are managed in the workplace is key. A workplace, in which these hazards are well-controlled, with an active culture of health and a supportive work environment, can enhance worker health and well-being, both on and off the job. Healthier employees result in fewer health claims, better safety records, and greater productivity.

Well-rested, alert employees are critical to safe and productive operations. Virtually everyone experiences some level of fatigue from time to time. However, excessive fatigue while working is an important condition in which the interrelationship of health, safety, and productivity can create a vicious or a virtuous cycle. Specific medical and lifestyle interventions have been shown to promote a well-rested and alert workforce. In addition, specific factors in the organization of work have been shown to promote either alertness or fatigue.

While a business cannot eliminate fatigue, it can effectively manage the associated risks. In fact, our comprehensive, data-driven Fatigue Risk Management Solution is a vital part of the new corporate toolkit. As well as forming part of a broader Safety Management System, it is the framework around which fatigue related risks are managed, operations are planned, and safety is prioritized.

Aim of the Thesis

The aim of our work was to redesign and improve upon the functionality of the iPhone application Crew Alert. As a premise for this thesis had some requirements of what they wanted. This included making Crew Alert into a universal application, keeping support for existing functionality and making a product that worked. Well in a cockpit environment as well as for everyday use.

2. CREW ALERT

Crew Alert is an iPhone application developed to increase the awareness of FRM amongst airline crew, mainly pilots. It does this by giving them a tool with which they can assess their own fatigue as well as learn about fatigue, how it works and what counter measures can be taken. With Crew Alert pilots know when an upcoming flight will be a hard flight (a flight where the pilot will be more fatigued than usual) and it can also help them prepare for such a flight.

Crew Alert works by calculating a prediction of the average fatigue for a population given a work and sleep schedule. Thus, it tells the user how tired an average individual is expected to be given its previous work and sleep schedule. The calculations are made with the Boeing Alertness Model (BAM), For upcoming flights, the model predicts the users sleep pattern based on user-settings to give a reasonable estimation of sleep prior to the flights

Crew Alert is meant to be used as a daily help, both as a reminder of the ever present fatigue aspect and to keep track of the user's schedule. At the start of each month the user preferably inserts their schedule, also known as a *roster*, automatically by import from the Roster Buster application. Roster Buster is a web service used by pilots to decode their PDF based schedule, or roster, to another format, like iCal, or importing it into Crew Alert. The users then daily enter when and how they slept. This is required from the user, to get the full benefit of the application. Crew Alert in turn offers its predictive functionality so that the user can compare flights and see which ones will be extra challenging.

Crew Alert also provides self-assessment tests for fatigue. With these, pilots can assess themselves in regard to the SP and KSS scales. This service can also be used along with Crew Alert's fatigue reporting feature, to give the pilot's airline a better understanding of their scheduling situation and the pressures put on their staff. The users can also anonymously upload their data in Crew Alert to increase the knowledge of sleep and fatigue. Once a year Crew Alert themselves hold a large scale data survey drive.

2.1 Description of Crewalert

Crew Alert, is a tab based application consisting of a *Graph-*, *Schedule-*, *Settings-* and *more* tab. The main tab is the *Graph* tab which contains the visualizations of the application's data; alertness, schedule and sleep.

The *Schedule* tab contains in large the same data but in a tabulated format. The *Settings* tab's content is quite self-explanatory, but some of the interesting settings are stated below. The *More* tab contains information such as legal documentation and an account section. Prominent features of the application are explained in the following sections.

2.2 Graph Tab

The *Graph* tab is the main venue of the application. Here the users finds a horizontal time line visualizing their predicted fatigue over time, in combination with their schedule, The view consists of two parts, the top part containing the schedule along with the sleep, and the bottom containing the alertness curve and the score of the users self-assessments. The whole view follows a horizontal time axis with vertical lines separating each day, each day is also indicated by its number in the month. At the top there is also a segmented control bar with which the user can choose the time interval display.

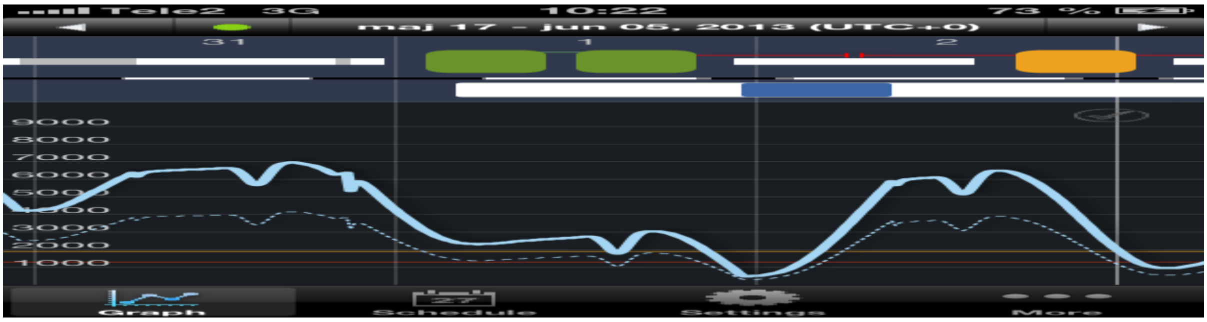


Fig. 1 The Graph tab in Crew Alert.

Alertness Curve

The bottom part of the *Graph* tab contains two curves, one solid and one dashed. These curves are plotted with the y-axis representing CAS points and the x-axis time, showing how the users CAS points go up and down as time passes by. The solid curve represents the average alertness of a population whilst the dashed curve represents the 90% confidence interval.

Along with the curve, two vertical lines (one amber and one red). These represent the alertness warning limits.

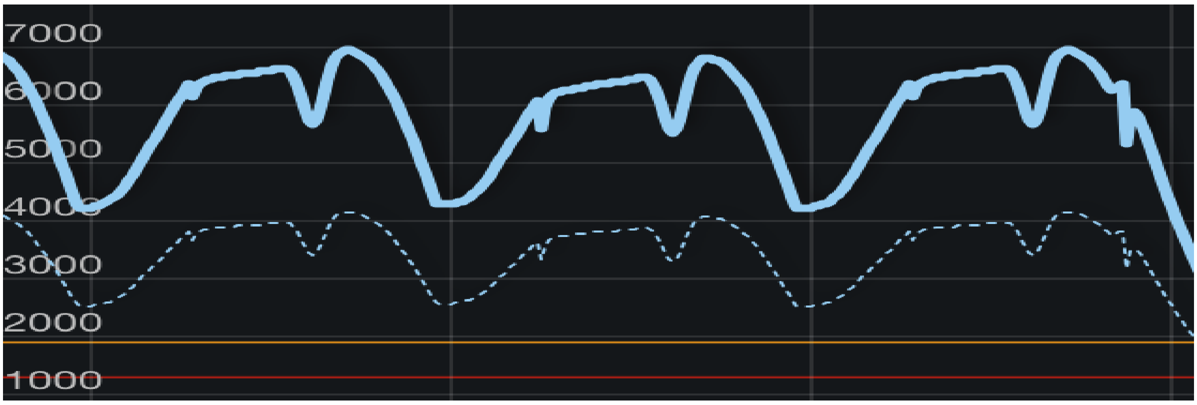


Fig. 2 Alertnescurve

Predicated Sleep/Awake Bar

The sleep/awake bar is a bar representing predicted sleep and sleep opportunities calculated by BAM, Gray areas indicate predicted sleep and white areas indicate a sleep opportunity. That is, a window of time where it is possible for the user to sleep. When no actual sleep data has been recorded by the user the white areas can be seen as predicted awake. When actual data has been inputted the gray areas disappear and only a white area is shown for that period of time.

Around duties the bar fades away due to the transfer times. Since a white area is a sleep opportunity it cannot stretch all the way to the duty since time is accounted for where the user should prepare for the work day and travel to the airport and thus cannot sleep. Inside duties there can be predicted sleep areas if the duty has in-flight sleep, so called controlled rest, set on it. When tapping a predicted sleep area, a tool tip box is displayed, showing start and stop times for the predicted sleep.

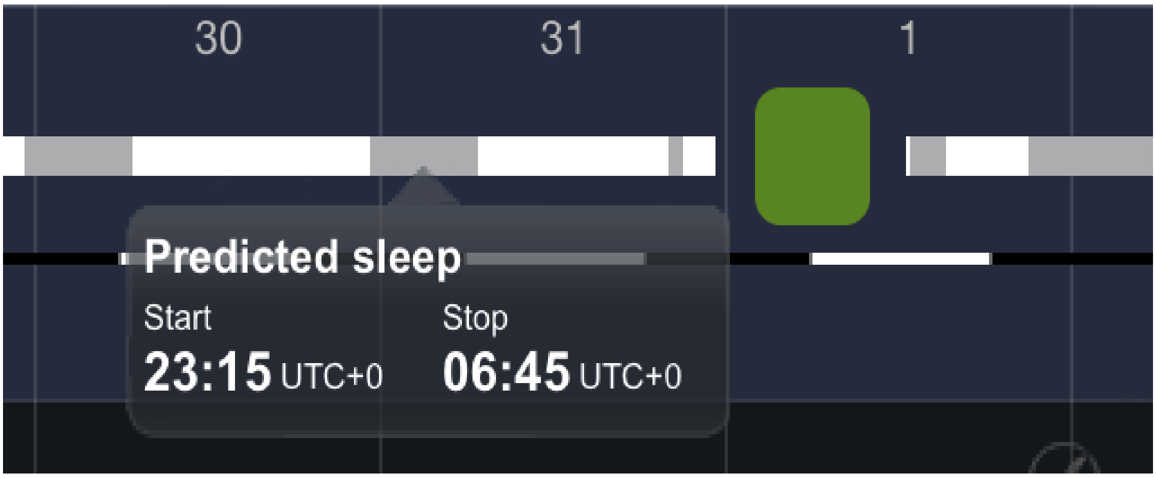


Fig.3 A Selected Predicated Sleep

Duties

Duties are represented as colored rectangles with rounded corners, showed on the predicted sleep/awake bar. The color indicates what type of duty it is. If the duty is a flight duty, the color also tells what alertness level of the user is predicted to have at top-of-decent for that flight. There are three levels for this; green, amber and red. Where green represents that everything is fine in terms of alertness. Whilst amber and red are two warning limits that can be edited by the user. These indicate, depending on what the user's own preferences are limits that tell the user if a flight will be harder than normal flights (amber) or an extremely hard flight (red).

If the duty is another type of duty, for instance a *Simulator Instruction* session or a *Ground Duty* it is colored with a fix color for that duty type. These duties do not change color by alertness. Between duties there can in some cases be either a red or a green line, the green line represents that the connected flights make up a larger duty period, a trip. Whilst red lines represent a mismatch in arrival and departure time zones.

That is, the user is not departing from the time zone he or she last arrived at. Somewhere the user had to have relocated without marking it in the schedule in Crew Alert with, for instance, a dead-head flight.

Fatigue Mitigation

Crew Alert has the functionality to calculate an optimal strategy for being as alert as possible at specified time. The user gives a time point at which they want to improve their alertness and crew alert presents the user with a screen exemplified, to induce the functionality the user needs to do a tap a hold interaction on the *Graph* view at a point in the future. One cannot make a strategy for a point in the past.

The user is presented with a set of *Do's* and *Don'ts* for eight different mitigation aspects. *Do's* are presented as a green bar and says that the user should *seek up* the aspect corresponding to this line for the time interval the bar represents in the *Graph* view.

Fig.4 Fatigue Mitigation

Don'ts work the same way but should now avoid the aspect; this is presented as a red bar. Tapping on a bar shows a tool tip box presenting the name of the aspect and the start and stop times. At the bottom of the screen there is a green or red number indicating how much alertness the user gains or loose with the given strategy, compared to the alertness if no strategy were used. To learn about an aspect the user can either tap the corresponding aspect icon or look it up under the More tab. Upon closing the Fatigue Mitigation functionality, done by pressing the red X button, the strategy for each aspect is lost and the user is left with a sleep/wake log with a red border, indicating that it was generated by the Fatigue Mitigation functionality.

3. OVERVIEW

Apple's iOS did not have an official name until the release of the iPhone software development kit (iPhone SDK) on March 6, 2008. Before then, Apple marketing literature simply stated that their iPhone runs a version of "OS X", a reference to iOS' parent operating system. When introduced, it was named iPhone OS. It was renamed iOS on June 7, 2010, as the iPhone was no longer the only device to run iOS. Apple licensed the "iOS" trademark from Cisco.

Versions

iPhone OS 1.x

- First iteration of Apple's touch-centric mobile operating system. No official name given on its initial release; Apple marketing literature simply stating the iPhone runs a version of Apple's desktop operating system, OS X. On March 6, 2008, with the release of the iPhone software development kit (iPhone SDK), Apple named it iPhone OS (they went on to rename it "iOS" on June 7, 2010) *iPhone OS 2.x*

- 2.0, the second major release of iOS, became available on July 11, 2008 with the release of the iPhone 3G. Devices running 1.x are upgradable to this version. This version of the OS introduces the App Store, making third-party applications available to the iPhone and iPod Touch.

iPhone OS 3.x

- 3.0 became available with the iPhone 3GS. It was released on June 17, 2009. This release added features such as copy and paste, and MMS. Not all features were available on the original iPhone. Devices running 2.x were upgradeable to this firmware.

iOS 4.x

- iOS 4 was made available to the public for the iPhone and iPod Touch on June 21, 2010, and is the first release to be renamed to simply "iOS". This is the first major iOS release to drop support for some devices. It is also the first major iOS release that iPod Touch users do not have to pay for.
- The iPhone 3G and iPod Touch (2nd generation) have limited features, including lack of multitasking capabilities and the ability to set a home screen wallpaper, while the iPhone, iPhone 3GS, iPod Touch (3rd & 4th generation) have all features enabled, such as multitasking. The iPhone and iPod Touch (1st generation) cannot run iOS 4.0 and above.

iOS 5.x

- iOS 5 was previewed to the public on June 6, 2011. It was released for iPhone 3GS, iPhone 4 (GSM and CDMA), iPhone 4S, iPod Touch (3rd & 4th generation), iPad, and iPad 2 on October 12, 2011.
- iOS 5.1.1 is the final release supported for the iPad (1st generation) and iPod Touch (3rd generation).

iOS 6.x

- iOS 6 was announced and previewed on June 11, 2012 during Apple Worldwide Developers Conference (WWDC) 2012, and its release was stated as Fall 2012. Following the pattern of previous iOS releases, some older devices were no longer supported, specifically the iPod Touch (3rd generation), and
- The iPad (1st generation). Supported devices include the iPhone 3GS and later; the iPod Touch (4th generation) and later; and the iPad 2 and later.[82]
- On September 12, 2012 at San Francisco's Yerba Buena Center for the Arts, among other items unveiled, Apple announced three iOS-related items: the next generation iPhone 5, the redesigned iPod Touch (5th generation), and the announcement of the release of iOS 6.0 the following week.
- iOS 6 was released to the public on September 19, 2012, through iTunes and over-the-air updates.
- iOS 6.1.6 is the final release supported for the iPhone 3GS and iPod Touch (4th generation).

iOS 7.x

- Apple announced iOS 7 on June 10, 2013 at its annual Apple Worldwide Developers Conference (WWDC) event, with release announced for sometime in Fall (Northern Hemisphere) or Spring (Southern Hemisphere) 2013. At their iPhone event on September 10, 2013, Apple announced the full release of iOS 7 for September 18, 2013, while also unveiling two new iPhone models: the iPhone 5C and iPhone 5S.
- With this release, support was once again dropped for older devices, specifically the iPhone 3GS and the iPod Touch (4th generation). Supported devices on this release include the iPhone 4 onwards, iPod Touch (5th generation), the iPad 2 onwards, and the iPad Mini (1st generation) onwards. iOS 7.1.1 is the final release for the iPhone 4

iPhone OS

- iPhone OS (known as OS X, or iPhone OS X in its early history) is a mobile operating system developed and marketed by Apple Inc. It is the default operating system of the iPhone, the iPod Touch, and the upcoming iPad. It is derived from Mac OS X, with which it shares the Darwin foundation, and is therefore a Unix-like operating system by nature. iPhone OS has four abstraction layers: the Core OS layer, the Core Services layer, the Media layer, and the Cocoa Touch layer.
- The operating system uses less than 500 megabytes of the device's memory.

- The **Core OS and Core Services layers** contain the fundamental interfaces for iPhone OS, including those used for accessing files, low-level data types, Bonjour services, network sockets, and so on. These interfaces are mostly C-based and include technologies such as Core Foundation, CF Network, SQLite, and access to POSIX threads and UNIX sockets among others.
- The **Media layer** contains the fundamental technologies used to support 2D and 3D drawing, audio, and video. This layer includes the C-based technologies OpenGL ES, Quartz, and Core Audio.
- The **Cocoa Touch layer** provides the fundamental infrastructure used by your application. For example, the Foundation framework provides object-oriented support for collections, file management, network operations, and more. It is based on Objective-C.

4. FRM IMPLEMENTATION

An FRMS is best employed within an organization's safety management system. This allows the risks associated with fatigue to be managed in a way similar to other hazards such as dangerous goods. An FRMS should be based on an internal risk assessment of the organization. This ensures that any fatigue management strategies being implemented are measured, appropriate, and targeted. There are several Canadian national standards for risk assessment, all of which clearly outline acceptable guidelines for risk management (e.g., CAN/CSA-Q850-97, CAN/CSA-Q634-91).

The fatigue risk management system described in this guide provides your company and employees with a recognized process based on likelihood and consequence and the need to identify, understand, and control the workplace hazard. The resources and time required for implementing a fatigue risk management system will be determined by the relative risk identified during your risk assessment process.

There are six major aspects to an FRMS:

1. Policies and Procedures

- Outline the commitment of organizational management to manage fatigue-related risk
- Detail the required procedures for managing fatigue at the operational level.

2. Responsibilities

- List personnel responsible for FRMS design, implementation, and maintenance
- Document responsibilities of individual employees and work groups

3. Risk Assessment/Management

- Scheduled versus actual hours of work
- Individual sleep patterns
- Symptom checklists
- Error/incident reporting

4. Training

- Promote knowledge in the workplace about risks, causes, and consequences of fatigue
- Ensure employees understand and can apply fatigue management strategies

5. Controls and Action Plans

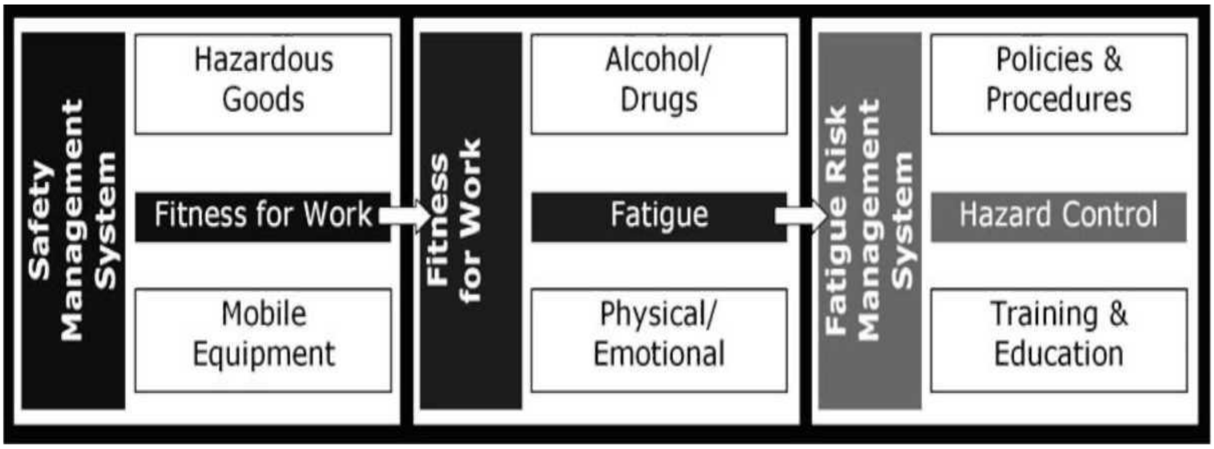
- Toolbox of methods used within the FRMS, including error reduction techniques ("fatigue proofing")
- Clear decision trees for managers and employees to use when fatigue has been identified as a risk

6. Audit and Review

- Documentation and data collection at regular intervals of how the FRMS works
- Review of the FRMS based on audit results.

The FRMS should be embedded within the existing SMS framework to allow fatigue to be managed within existing organizational safety structures. This also ensures that responsibility for managing fatigue risk is shared between employer and employee.

It may also allow safety professionals or other stakeholders in the company to develop a cost-effective FRMS without needing to call in outside fatigue expertise. However, it is important to have an understanding and appreciation of fatigue-related risk within a workplace. The figure below illustrates how fatigue can be incorporated into an overarching SMS.



Fatigue Risk Management Systems Work Best within the Framework of a Larger Safety Management System

Fatigue Detection Using Smart Phones

These traffic-related deaths and financial losses have encouraged the development of technologies to mitigate the risks of driver fatigue. Researchers often used vehicle-mounted cameras and laptop computers to detect and monitor driver fatigue. Indicators of fatigue include the movement of driver's face, eye blink rate, head nod, and yawn of mouth etc. Another approach is to use electrophysiological signals to detect driver fatigue. The relative powers of EEG signals, for example, (a+p)/9, a/ p, (9+ a)/ (a+p) and 9/ p, are indicative of driver fatigue.

A more technically feasible and financially practical solution is to build the fatigue detection technology using smartphones. As of 2012, as many as 45% of American adults owned a smartphone. The computational powers of smartphones keep increasing, which allows the computer vision algorithms to run reasonably fast in smartphones.

A smartphone-based fatigue detection technology would be more portable and affordable than many alternative fatigue detection systems, which use devoted in-vehicle cameras or EEG sensors.

The proposed system in this paper uses a smartphone (either an Android smartphone or an iPhone) as a driver fatigue detection system. The front camera of a smartphone captures images of the driver, and then feeds the images to the CPU of the smartphone for image processing. Intel's open-source computer vision framework, the Open CV 2.3 for the Android and iOS operating system, is used to develop the computer vision algorithms for face detection and eye detection.

5. CONCEPT DESIGN

The concept design started in trying to facilitate a broader use of the application, giving more substance to the user, to make him come back time and time again. Our idea for this was to have Crew Alert as a planning and preparation tool. Before beginning a trip the pilot would use Crew Alert to visualize the trip and mentally prepare for it. Thus incorporating Crew Alert into the pilots' workday.

Lowering Of Learning Curve

Semi parallel to the concept design, work was also put into lowering the complexity of the existing features that were reused, making it more intuitive, benefiting both old and new users. Both from the survey and our own experience learning and exploring the application, two aspects of the application were identified as extra problematic. One was to log data and one was the schedule part of the *Graph* tab, these were seen as high priority design targets.

Schedule

The reworking of the time-line of the schedule of the *Graph* tab was of some concern during the design. This would help the user both learn the application faster and to feel more at home in the application. One problem that it suffers is the double way sleep is displayed, with one bar for predicted sleep and one for actual sleep.

As a intuitive way of simplifying this, these two bars were merged into one. To not confuse the user as to what sleep he had logged and what sleep the model had predicted, the look of the different sleeps had to be reworked to match visually. The confirmed sleeps were given a paler color, to match the predicted sleeps, which now became neighbors. They were also given a black border to make them feel more solid compared to the predicted (unconfirmed) sleep.

Moving the actual sleep up to the predicted sleep bar left the light bar "hanging" on the bottom. To rebalance the design it was instead placed above the new combined sleep log. To make the day night more visible, the light bar now casts a translucent dark/light layer onto the entire *Graph* view. Since light exposure is a important factor of fatigue, to compare the alertness curve and schedule to the daylight in a easy way is of great value.

Sleep Logging

The matter of sleep logging is of great importance since it is one of the more interaction heavy activities in the application, except perhaps inputting your schedule which could be imported automatically. This means that ease of use here is important since it is something you need to do from scratch, as opposed to the more specialized, to get the benefit of the application. This means making this system better benefits both of the main design goals of bringing the user back recurrently and making the application easier to learn.

The current system had unnecessary complexity attached to it, following more of an implementation model than that the user's mental model. Many interventions were taken to aid this process. The first step was, along with the reworking of the timeline, to collapse a number of diary screens into one, making the addition of sleep, wake and duties all done in one place.

Another big change was to remove the need for a sleep/wake log. It should be noted that sleep is logged as usual; the difference is that sleep can be logged without a sleep/wake log to put it in. Instead a system was worked out, where the user only entered sleep, and the actual awake was automatically generated by the application. The system also lets the user correct the assumptions and instead input the awake (or sleep) manually.

Alarm Clock

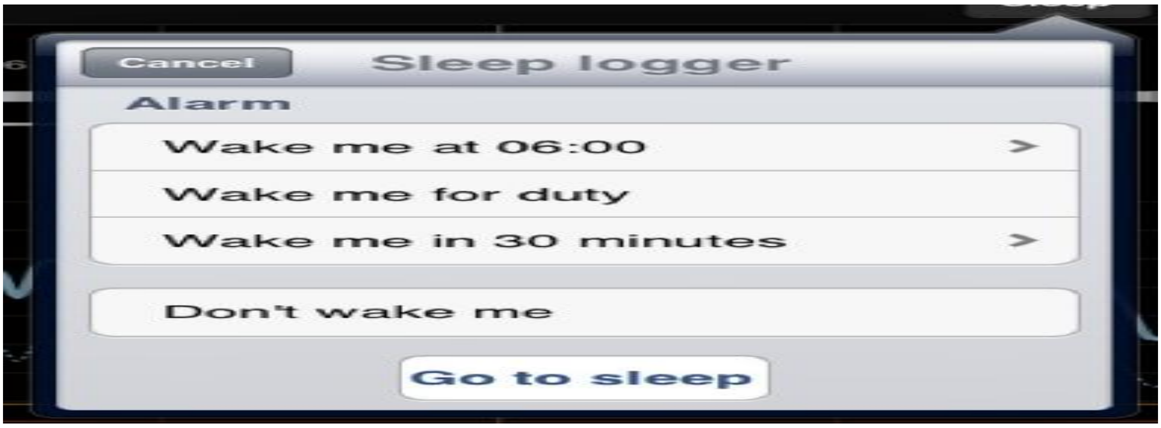
In order to help the user log their sleep, an alarm clock system was designed. This would allow the users to do their sleep logging implicitly while using the alarm clock feature. The alarm clock work similarly to sleep cycle-recognizing alarms, where you activate it when you go to sleep, and deactivate it by turning off the alarm in the morning. The system would then automatically add a sleep log for the time interval.

Apart from being a useful feature, to have it alongside the other features of Crew Alert could make it more time effective to use. Since the user might want to check other information in Crew Alert at the same time as setting the alarm. For instance when they start in the morning and where to they are going to fly, to prepare for the next day.

This would make up another reason for pilots to use the application. If he for instance already use some sort of sleep logger, alarm clock or combination of the two, this would give the same functionality with the extra benefit of the alertness prediction.

A possible addition to this system is a Wake me for duty option. This would allow the airline the user is working for, to remotely postpone the alarm, if it turns out that the first flight for the sleeping user is going to be delayed. Thus not waking him before actually necessary.

This feature might sound strange, but it is a service already used by airlines when the pilot is sleeping at a hotel on an out-station. Then the airline has the opportunity to notify the hotel staff about when to place the wake-up call. Although the system looked promising in its feature set as well as being a requested functionality by users, we did not have enough time to implement it.



Alarm Clock

Local Time Concept

To make it easier to input time zones a *Local Time* option, referred to as introduced. The LT option differs from what one might think, it does not depend on where you are at the time. It rather depends on where you will be at the time of the entry you edit (or create). For instance, if the user inputs airport first and then time, when adding a duty, the application calculates the time zone to that of the selected airport.

This distinction was made, in part, to accommodate for schedules given in local time, for each airport, as for instance SAS provides their schedules. In the case that the user fills the flights in manually he does not have to switch the time zone for each take-off and landing. Even though we hope that as many as possible use automatic import of the schedule, and therefore does not need this feature, we want to provide ease of use to those who do not as well. The editing of imported flights after rescheduling, and so on, could be an issue where the filling in or changing of time zones could be helped by this feature.

Trip Bar

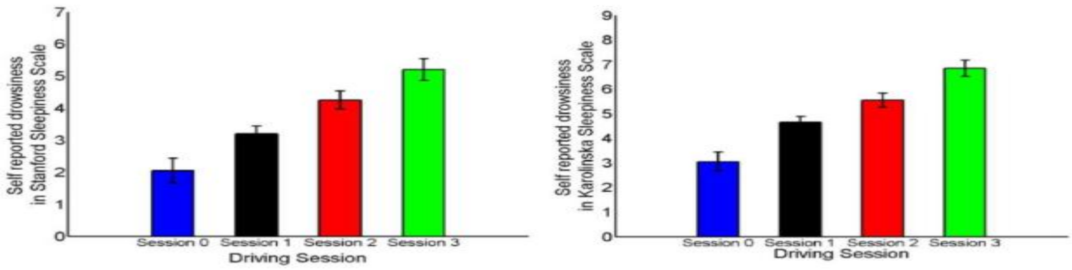
This bar has its foundation in the concept of Crew Alert as a planning tool. It was inspired by the bottom bar in Apple's calendar application for the iPod; the bar was at creation given the ability to show days, trips and months. The Crew Alert application Terminal Charts also gave some inspiration towards having a bottom bar.

The Trip Bar, seen at the bottom of image, evolved from the idea of a scrollbar that visualized the flights and fatigue, like a miniature version of the *Graph.* Bar for a early mockup. This idea was however left due to its relative complicity, for the iCal inspired version, which would be more recognizable to iPad users.

6. SELF-REPORTED DROWSINESS

At the beginning and ending of each session, subjects self-reported their drowsiness in two scales, the Stanford Sleepiness Scale (SSS) and Karolina Sleepiness Scale (KSS). As the self-reported drowsiness steadily increases across drive sessions in both scales.

The self-reported drowsiness score in SSS produces a main effect of drive sessions, $F(3, 57) = 60.17$, $p<.001$, $q^2= 0.76$. Pair-wise comparisons show that all the session-pairs are significantly different from each other, all $ps<0.01$. The data of self-reported drowsiness score in the KSS produces similar results. Self-reported drowsiness data thus confirm the successful manipulation of the driving study to incur drowsiness.



Self-Reported Drowsiness across Drivers Sessions

Application Features

- Crew Alert provides the functionality of much more expensive alertness modeling products at a fraction of the cost.
- As flight and duty time limits increasingly move towards Fatigue Risk Management Systems, tools like Crew Alert will become increasingly useful for managing fatigue in any airline operation.
- While Crew Alert is designed with flight deck and cabin crews in mind, it is applicable to any 24/7 operation where human performance is crucial to safety.

7. CONCLUSION

The intended accomplishment with this thesis was to explore the porting, designing and usability work of an application running on iOS and used by pilots. It was of importance to get the application at hand running smoothly on a variety of devices, with different screen sizes, resolutions, orientations and form-factors. To get accustomed to the preexisting code, and do the basic porting code and scaling of interfaces and graphics also took much more time than expected. The work was too large to thoroughly examine and make well motivated design decisions for each part of the application. The implementation could also be down prioritized in favor of more in depth design work.

REFERENCES

[1] Crew Alert, Fatigue Risk Management (Feb 2013) URL http://www.Crew Alert.com/firm.

[2] D.Hellerstrom, H.Eriksson, E.Rooming, T.Klemets, the Best Rest, “Aero Safety World Magazine”.

[3] C.H.OsterLund, J.Wildlund, “A Mobile Application for Flight Safety”: The Crew Alert, Master’s thesis Chalmers University of Technology and University of Gothenburg(2010).

[4] Boeing, about us, Boeing in brief (Jun 2013).URL: http://www.boeing.com/boeing/ompanyoffices/ abouts/brief.page.

[5] Crew Alert, Crew Alert homepage (Jun 2013). URL http://www. Crew Alart.com/.

[6] Roster Buster, Roster Buster (Jun 2013). URL http://www.rosterbuster.com.

[7] FAA, Congress bill (Feb 2013) URL: http://www.faa.gov/other_visit/aviation_industry/airlineoperators/ airlinesafety/info/allinfo/media/age65_bill.pdf.

[8] S.B. Mushin, Effects of age and Expertise on pilot performance, Journal Watch Neurology http://Search.proquest.com/docview/1284339917?accountid=10041.

[9] M.Ziefle, U.Schroeder, J.Streak, T.Michel How Younger and older adults master the usage of hyperlinks in small screen devices, in: proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’07, ACM, New York, NY, USA, 2007, pp.307-316. URL http://doi.acm.org/10.1145/1240624.1240676.

[10] Merritt, Replicating Hosted: “A Study of pilots in eighteen countries, University oftexas”.1997.

[11] FAA, Medical certificate requirements (Feb 2013).URL http://www.faa.gov/pilots/become/medical/.

[12] Flight Physical, Second class vision standards for pilots(Feb 2013).URL http://flightphysical.com/part67/class2/67subc_67203.htm.

[13] D.Vogel, R.Balakrishnan, Occlusion-aware interfaces, in :proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI’10, ACM, New York, NY, USA 2010,PP.263-272.URL http://doi.acm.org.proxy.lib.chalmers.se/10.1145/1753326.1753365.

[14] P.Pombinho, Information visualization on mobile environments, in: Proceedings of the 12th international conference on Human Computer interaction with mobile devices and services, Mobile HCI’10, ACM, New York ,NY, USA, 2010,pp.493-494 URLhttp://doi.acm.org/10.1145/1851600.1851722.

[15] T.A.C. Reis, Improving mobile interaction with context-awareness, multimodality, and adaptive interfaces. In: Proceedings of the 12th international conference on Human computer interaction with mobile devices and services, Mobile HCI’10, ACM, New York, NY, USA, 2010, PP.495-496. URL http://doi.acm.org/ 10.1145/1851600.1851723.

[16] T. Akerstedt, R.Mallard, A.Samuel, M.Simons, M.Spencer, in: Meeting to discuss the role of EU FTL legislation in reducing cumulative fatigue in civil aviation, ETSC, 2003. URL: http://www.eurocockpit.be/sites/default/files/Akerstedt- Mollard-Samel-Simons-2003.pdf.

[17] I.IATA, ICAO, Fatigue Risk Management Systems –Implementation Guide for Operators, IATA, 2011. URL: http://www.iata.org/publications/Documents/FRMS%20Implementation%20Guide%20 for %20operators%201st %20Edition%20English.pdf.

[18] Ake Albert, D.Hellerstrom, T.Clemet's, "A Comprehensive Investigation of Flight and Duty Time Limitations and their Ability to Control Crew Fatigue", Crew Alert, 2011.

[19] Crew Alert, The Boeing Alertness model (Feb2013). URL: http://www1.Crew Alert.com/documents/aviation/commercial/BAM_IV.pdf.