CHALMERS

Mobile Applications Design in Fatigue Risk Management

Alertness on the fly.

ANTON PERSSON
JOHANNES ANDERSSON

Interaction Design and Technologies
Chalmers University of Technology
Gothenburg, Sweden 2013
Master's Thesis 2013:127

Mobile Applications Design in Fatigue Risk Management

JOHANNES ANDERSSON
ANTON PERSSON

Examiner: STAFFAN BJÖRK

ISSN: 1651-4769 Technical report no: 2013:127 Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone +46 (0)31-772 1000

Gothenburg 2013

Abstract

This report describes the redesign and development of the iPhone application CrewAlert. 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 every day use.

Acknowledgements

For giving us this great opportunity to expand our views on both design and coding we would like to thank Jeppesen, and most of all our supervisor Tomas Klemets. We would also like to thank the rest of the team involved in our endeavor; Christian Holth Österlund, David Hellsertröm and David Karlsson.

We would also like to thank our supervisor at Chalmers, Olof Torgersson, for helping us to keep our eyes on the goal.

Contents

Glossary

Agile development A way of working when developing software, the focus being on not locking what is to be done but instead be able to change the plan quickly. i

BAM Boeing Alertness Model, alertness prediction model developed by Boeing and Jeppesen. i, 4, 7, 16, 17

Brainstorming A method use to generate ideas, for instance during design, see section 4.2. i, 24

CAS Common Alertness Scale, alertness scale put forward by Jeppesen and Boeing. i, 16

Circadian Rhythm The biological cycle which regulates sleep pattern, among other things. Could be described as the biological clock. i

Crunch A development slang indicating long hours with quick development cycles at the end of a project. i, 68

CSFM Context Sensitive Fatigue Mitigation, a feature in CrewAlert that enables the user to get guidelines on how to increase their alertness at a certain time. See section 2.3.2 i

Dead-head When a pilot flies as a passenger in order to get to the airport that his or her next duty starts from. i, 8

Duty A work period consisting of one or more flights. i

Extreme Programming An agile development method, commonly based on short iterations and continuous testing with simplicity being key. i

FAA The Federal Aviation Administration i, 5

Fatigue Mental or physical weariness due to lack of sleep. i

FRM Fatigue Risk Management i, 1, 4, 12, 23, 26, 70, 80

FRMS Fatigue Risk Management System i, 3, 16, 17

FTL Flight Time Limitations, rules and regulation for flight safety on, among other things, how long a pilot can be flying without rest. i, 3, 15, 16, 21, 22

GDD Goal-Directed Design i, 68

Hard flight Here used to describe a flight where the pilot will be more fatigued than usual. i, 4

HCD Human Centered Design i, 24

HSL Habitual Sleep Length, the time one usually sleeps without external awakening from an alarm or such. i, 17, 42

ICAO International Civil Aviation Organization i

IDE Integrated Development Environment i, 23

IOS The operating system used by Apple on their mobile devices; iPhone, iPad and iPod touch i, 23, 66

KSS Karolinska Sleepiness Scale, describing sleepiness on a scale from 1 to 9. i, 4, 16

Leg One unit of flight, part of a duty. i

Mercurial A distributed version control software. i, 23

Persona A design construct used to get a better understanding of the intended user of a product, see section 4.4. i, 24, 25

Photoshop A powerful image editing and creating software. i, 25

Roster A pilot's work schedule as given by the airline i, 21

Scrum Development framework. [28] i, 3, 68

Sleep debt Time of sleep deviated from one's habitual sleep length. i

Sleep inertia The increased fatigue experienced temporarily upon awakening. i, 41

SP Samn-Perelli Scale, describing sleepiness on a scale from 1 to 7. i, 4, 16

Sprint A time frame in the Scrum development framework. Could be seen as a mini project within the project. i, 23

Top of decent Or TOD, the time a pilot starts the descent prior to landing. i

Trip A chain of flights that starts and ends at the same location. Can span over several days. i, 8

Universal application iOS application running on all iOS devices. i

Version control A software used to keep track of previous version of the code base in a development project. Giving an easy way to collaborate on the code as well as the ability to go back in time if needed. i, 23

Xcode IDE created by Apple used for iOS software development. i, 23, 25

XP Extreme Programming i, 68

Zulu time The time zone UTC +/- 0. i

1

Introduction

Fatigue Risk Management (FRM) is a novel take on otherwise quite fastidious laws on the working hours of commercial airline pilots.[1] A leading actor in the field of FRM is Jeppesen, a company working with, among other things, flight schedules. With FRM Jeppesen helps consolidating and improving their flight schedules by using a scientific model to predict and decrease fatigue, but also help remedy the effects of it. FRM is meant to help flight crew to manage the risks that comes along with long hours in a better way than counting the minutes from when the pilot last signed out from work. FRM provides a way of upholding the related safety in a way more suited to the way we as humans actually work.[2]

As an initiative by Jeppesen, who have developed an iPhone application, CrewAlert, relating to FRM as part of a previous masters thesis, this project was started.[3] The task at hand was to redesign the previous application, giving the crew members the ability to use both the iPad and the iPhone, as well as make a more usable and streamlined experience for the user.

Today CrewAlert functions as a sleep-diary that along with your work schedule predicts your alertness for any given time, and presents it visually. This enables the user to see what flights are more challenging from a fatigue perspective.

1.1 Aim

The aim of our work was to redesign and improve upon the functionality of the iPhone application CrewAlert by Jeppesen. As a premise for this thesis Jeppesen had some requirements of what they wanted. This included making CrewAlert 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. Given this, we proposed the following research question.

> *How to redesign and extend the functionality of the CrewAlert application by Jeppesen in order to improve its usability, usefulness and cross format compatibility?*

As analysis of the product progressed, we found the following two focus points. For information on the reasoning behind these see section 5.1.1.

> *How to improve the interface and interactions in order to accommodate for new users?*

> *How to improve the longevity of the application, providing substance for the users after the novelty wears off?*

2

Background

Boeing, founded in 1916, is the world's largest aerospace company and leading manufacturer of commercial jetliners and defense, space and security systems, with 170'000 employees spanning over 70 countries.[4] A subsidiary to them is Jeppesen, a company founded in 1934 with 3200 employees. Jeppesen focuses on navigational information, operations management and optimization solutions.[5] In 2010 a master thesis was performed at their Gothenburg office, focusing on crew management, which gave raise to the iOS application CrewAlert as a part of their Fatigue Risk Management work.[3]

Apart from helping to increase the awareness of fatigue, CrewAlert is also a help in setting up Fatigue Risk Management Systems (FRMSs). A FRMS is, as the name hints, a system to prevent and handle fatigue within an airline. A FRMS can take many forms, and are implemented by the airlines themselves. If an airline's FRMS is approved by the necessary regulatory organs, the airline can then move away from the regular FTLs which are the safety rules governing if a pilot is allowed to fly or not given his previous schedule.

CrewAlert is helpful in a number of ways in creating this type of system, and is in the future thought to be a cornerstone in Jeppesen's work with FRMSs. Its main strength is that it is easily distributed to the airline staff, all that is needed is an iPhone. CrewAlert can then help the airline collect a number of metrics from its staff that is needed for their FRMS. These include fatigue reports, alertness scorings, sleep logs and other feedback, dependent on how the airline's FRMS is designed.

Since then development has given raise to the *Electronic flight bag*. This is a digitalization of the paper document and maps, weighing up to 20 kg, that the pilots need to bring with them on a flight. On some airlines the device chosen to replace this bag is the Apple iPad. This has given the need to expand the functionality of CrewAlert to support iPad devices as well as an updated look and feel to fit the new cockpit environment.

2.1 Purpose of CrewAlert

CrewAlert is an iPhone application developed by Jeppesen which aims to increase the awareness of FRM amongst airline crew, mainly pilots. It does this by giving them a tool with which they can asses their own fatigue as well as learn about fatigue, how it works and what counter measures can be taken. With CrewAlert 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.

CrewAlert 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), developed by Jeppesen and Boeing. 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. The result of the prediction is given in alertness units on the Common Alertness Scale, *CAS*, a scale from 0 to 10'000, also put forward by Jeppesen and Boeing.

CrewAlert 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 users schedule. At the start of each month the user preferably inserts their schedule, also known as a *roster*, automatically by import from the RosterBuster application. RosterBuster is a web service used by pilots to decode their PDF based schedule, or roster, to another format, like iCal, or importing it into CrewAlert.[6] The users then daily enters when and how they slept. This is required from the user, to get the full benefit of the application. CrewAlert in turn offers its predictive functionality so that the user can compare flights and see which ones will be extra challenging.

CrewAlert 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 CrewAlert'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 CrewAlert to Jeppesen to increase the knowledge of sleep and fatigue. Once a year Jeppesen themselves hold a large scale data survey drive.

Parties that benefit from this application are Jeppesen, airlines, the user and the scientific community; both in regard to flight safety and sleep science. The increased knowledge on sleep is used to further improve the the alertness model in the application. The application will also benefit the development of Flight Time Limitations, FTLs, and help regulatory organs, with its data gathering and predicting capabilities.

2.2 Users of CrewAlert

The intended user group for this application is commercial airline pilots. The defacto standard retirement age is 65, as The Federal Aviation Administration (FAA) recently changed it from 60, which was already the case in most of other countries.[7, 8] This loosening of age restriction puts older users in our target group to a larger extent, they will according to Ziefle et al. have a harder time orienting themselves in the application. However Merritt indicates that pilots are more interested in technology than the general public which will give them expert domain knowledge which according to Ziefle et al. compensates for the age.[9, 10]

All commercial pilots must have a class two medical license which does not allow for color blindness.[11, 12] Thus we do not have to account for color blind people in our design.

2.3 Description of CrewAlert

In this section the CrewAlert application is described as it was prior to the start of the thesis. For further information about CrewAlert we recommend the thesis project by Österlund and Widlund, which originally created the application.[3]

CrewAlert 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.

2.3.1 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, see figure 2.1. 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 an 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 displayed.

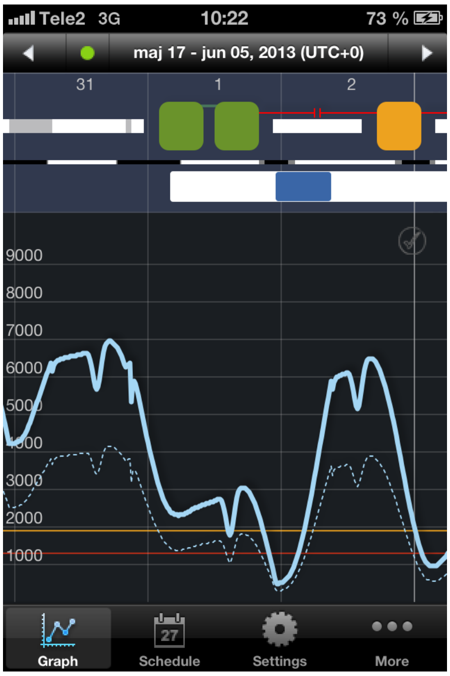


Figure 2.1: The Graph tab in CrewAlert.

Alertness curve

The bottom part of the *Graph* tab, shown in figure 2.2, 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) are shown. These represent the alertness warning limits.

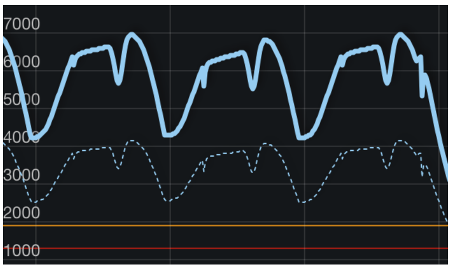
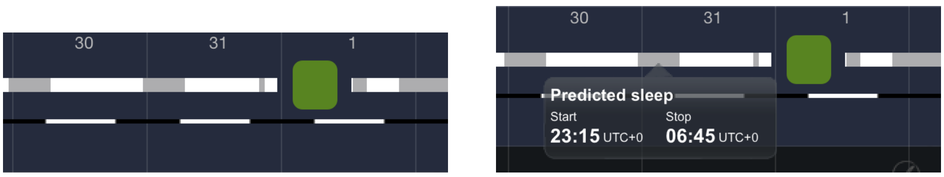


Figure 2.2: The alertness curves.

Schedule

The top part of the *Graph* tab shows the schedule, which consists of: a predicted sleep/awake bar, the user's duties, user's sleep journals and a light bar. Each of which will be explained below.

Predicted sleep/awake bar The sleep/awake bar is a bar representing predicted sleep and sleep opportunities calculated by BAM, see figure 2.3(a). 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 can not 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 can not 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, as in figure 2.3(b), showing start and stop times for the predicted sleep.

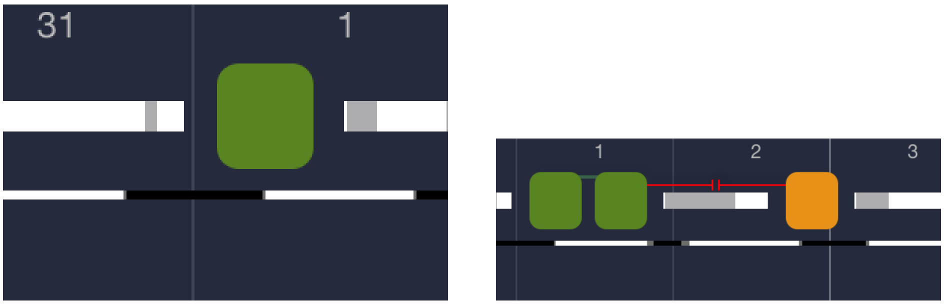


(a) (b)

Figure 2.3: The predicted sleep/awake bar. With (b) showing a selected predicted sleep.

Duties Duties are represented as colored rectangles with rounded corners, see figure 2.4(a) 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 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 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, as in figure 2.4(b). 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 CrewAlert with, for instance, a dead-head flight.



(a) (b)

Figure 2.4: Duties without and with lines between them.

Upon tapping a duty it expands down to the solid alertness curve giving a comparison on how the alertness will be during the flight, as seen in figure 2.5. To the left of the duty a tool tip box is displayed showing information about the duty. This includes departure and arrival airports and times, the time zone for these times, how many legs the duty consists of, the shift in time zones, the CAS score at top-of-decent and a comment on the flight, if the user enters one.

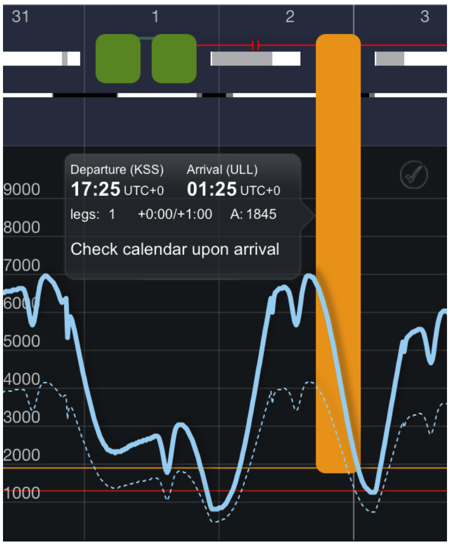


Figure 2.5: An expanded duty, showing it's tool-tip. The alertness goes beneath the amber limit, giving it a amber color.

Sleep/awake bar based on user data At the bottom of the schedule is another sleep/awake bar that shows user's logged sleep and awake. These are logs with a start and end time, during this period the user is considered to be awake, if nothing else is entered. Inside of these logs the user can add a sleep period. Logs are colored white and sleeps are colored blue, as in figure 2.6. Upon tapping on a sleep period a tool tip box is displayed showing the start and end time of the sleep.

Figure 2.6: The user based sleep/awake bar.

Light bar In between the predicted and user's sleep/awake bars there is a light bar, shown in figure 2.7. This bar shows the times the sun is up (white color) and down (black color) for the area the user is in at that time. Dusk and dawn are also shown (gray color) This is calculated with the help of the departure and arrival airports entered for the duties.

Figure 2.7: The light bar.

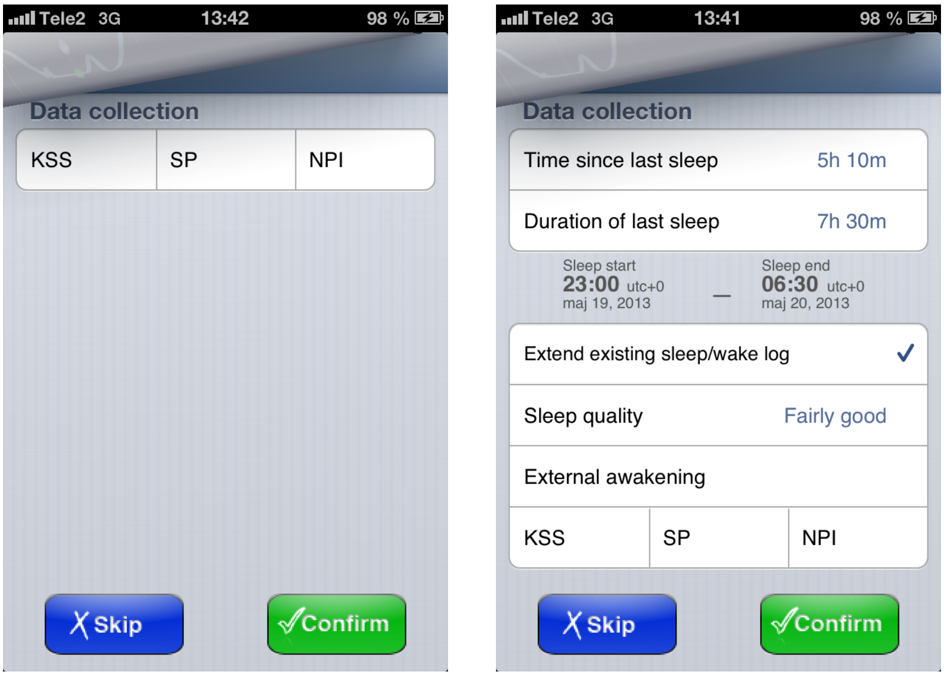
2.3.2 Feature set

Apart from the Graph view tab, CrewAlert also contains a number of prevalent features. The most important of these are explained as follows.

Data Collection Screen

The driving part in the logging of data in CrewAlert is the *Data Collection Screen*. In this view the user can enter self assessments of their alertness and log their last sleep. The screen is spawned automatically when opening CrewAlert, if a certain interval has passed since the user last opened the application. The user can adjust this interval in the *Settings* tab.

The *Data Collection Screen* will be the first thing the user is presented with upon launching CrewAlert. The view can also be brought up by the user by double-tapping the Data Collection icon in the *Graph* view (can be seen in the upper-right corner of the graph view in fig. 2.5). By default the screen only asks the user to make self assessments on alertness, as seen in figure 2.8(a). If, however, the view is displayed a set time after CrewAlert predicted that the user has slept it will also ask for information about this sleep, as seen in figure 2.8(b). By confirming a *Data Collection Screen* which asks for sleeps it will automatically create a new *sleep/wake log* for the given interval and fill it will sleep. If the option *Extend existing sleep/wake log* is checked it will also extend the last *sleep/wake log* with awake up to the start of this log (this option only exists if not more than a set amount of hours have passed since the last log).



(a) (b)

Figure 2.8: The *Data Collection Screen* with and without the sleep input form.

Scenarios

The user can have multiple setups of their work schedule and sleep pattern, these are called scenarios, see figure 2.9. With these the user can simulate different strategies and alternative work hours to see how this affect fatigue levels. The scenario screen can be found by clicking the scenario button, in the segment controller of the Graph view. When in the main scenario the button is displaying a green dot, when in another scenario it displays a red dot. This is to warn the user that the application is in a scenario.

Fatigue Mitigation

CrewAlert has the functionality to calculate an optimal strategy for being as alert as possible at a specified time. The user gives a time point at which they want to improve their alertness and CrewAlert presents the user with a screen exemplified in figure 2.10.

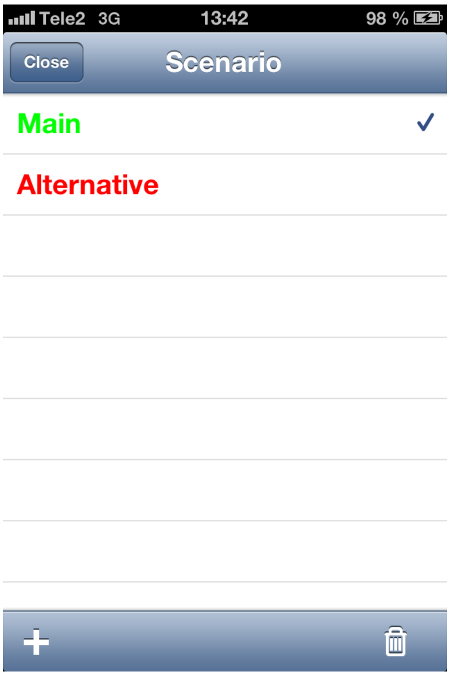


Figure 2.9: The Scenario screen.

To induce the functionality the user needs to do a tap an hold interaction on the *Graph* view at a point in the future. One can not 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. *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.

Uploading data

In order to expand the knowledge of fatigue the user can upload their CrewAlert data to Jeppesen which in turn makes it available to the research community. This is done by creating an account inside the CrewAlert application and using the *Update and Share* feature in the *More* tab, see figure 2.11(a). When uploading the user decides which time

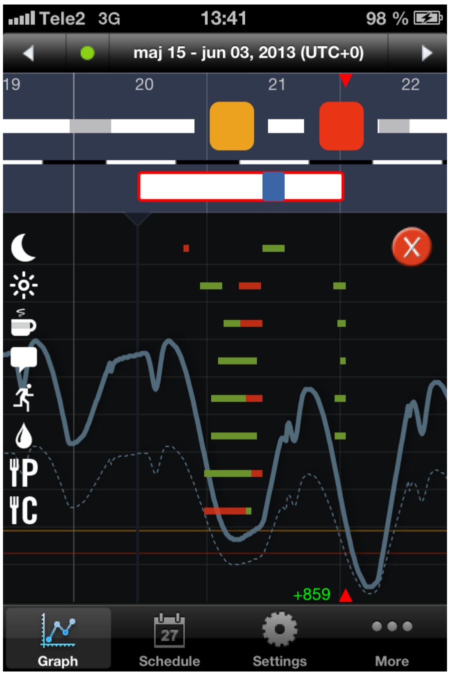


Figure 2.10: The fatigue mitigation screen showing the different aspects to seek up and avoid, along with a generated suggested sleep log.

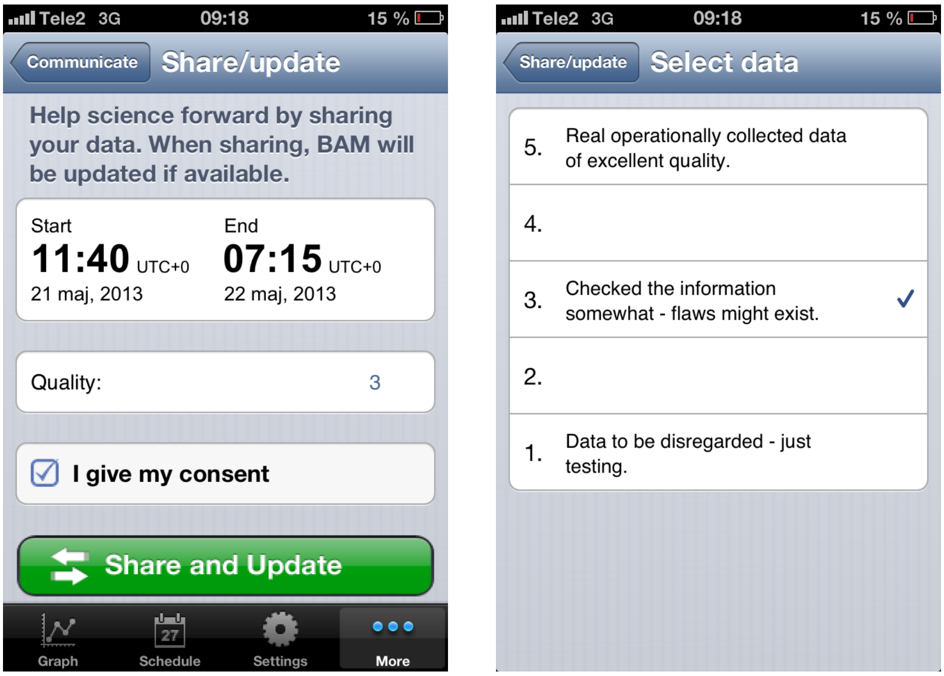
period of data to send and also provide information about the quality of the data, how close to reality it is. Upon uploading data the user gets an updated version of BAM, if available, as a motivation to share his data.

Fatigue Reporting

The user can generate reports when they experience duties that are alarmingly fatiguing, or otherwise interesting from a FRM perspective. A set of questions trying to target the cause and effect of the incident are asked. The report can be sent by email to the airline for which the user is working for so that they can improve their scheduling or take other precautions.

Roster import

If the user uses the RosterBuster service, they can import their schedule into CrewAlert by tapping a specific link provided by RosterBuster. This link will open CrewAlert which will then import the data.



(a) (b)

Figure 2.11: The data upload section.

2.3.3 Settings and More tabs

The application contains two more tabs, *Settings* and *More*. The application settings is used for variety of things, some of which are more important for the functionality of the application, some being more personal used during data upload. The important features of these tabs are explained below.

Time zones

Three settings exist for choosing time zones. The first is the *Default time base*; this setting is used to set what time zone the application assumes you will enter times in. The second setting is that of your *Home base time zone*, and is used to determine in what time zone the user will be when no prior flights have been entered into the application's schedule. The final setting is the *Graph view time base*. This setting determines the time zone the *Graph* tab will be showed in.

Warning limits

Throughout CrewAlert there are two warning limits being used, amber and red. These are used on duties to indicate that this duty will be harder than regular. The limits are personal, and should be set to levels matching the users preferences. In the *Settings* tab the user can set at which CAS points these limits will be.

Transfer times

In order for the model to calculate predicted sleep and awake it needs to know how long it typically takes for the user to go from awakening until they have arrived at work and from ending work until they can be asleep. These times can be set in the *Settings* tab.

Habitual sleep length

Here the user sets the time he usually sleeps, without external awakening. This setting is used to personalize the sleep prediction, it is also of interest during data collection.

Survey Mode

This setting is used to decrease the bias from the user when doing data surveys. This is done by removing the alertness curve in the *Graph* view, so that the user can not willingly, or not, try to align their data collection tests with the alertness curve.

3

Theory

When designing for mobile and touch-based devices there are many caveats to take into account; the screen real-estate, the occlusion of interfaces during touch, the varying contexts of use to name a few.[13, 14] Though the use contexts may be diverse, this challenge can also be seen as a way of giving new and better functionality to the device.[15]

3.1 Fatigue Risk Management

Fatigue plays an important role in explaining human errors within the pilot profession and a substantial amount of these error stem from fatigue. According to Akerstedt et. al. one in six aviation accidents can be attributed to fatigue. [16] The statistics of fatigue related accidents can not, of course, be said to be exact, but it shows the importance of managing the risk of fatigue.

Fatigue Risk Management as a safety approach is starting to gain momentum within the aviation industry. The idea is that rather than fixed times for rest in-between duties or flights (FTL) fatigue level is assessed, and based upon this level a pilots capabilities is judged. One component of FRM is the understanding of fatigue: What is causing it, how it effects people and what can be done to decrease the fatigue. Another is the industry regulations managing the fatigue aspect: How should routines, guidelines and responsibilities be formulated to minimize both the fatigue and its effect.

3.1.1 Fatigue in aviation

As a basis for discussing fatigue, the definition from the document *Fatigue Risk Management Systems - Implementation Guide for Operators* is used. This is a joint document from IATA, IFALPA and ICAO to help airlines adapt the FRM model.

> *"[Fatigue is] 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 crew member's alertness and ability to safely operate an aircraft or perform safety related duties."* [17]

For a pilot to fall asleep, or make faulty decisions on the job due to fatigue could, of course, be very harmful and even deadly. Besides the dangers induced when being fatigued they are also subject to being more fatigued than the average person due to their specific working conditions. These factors include common ones such as long hours, irregular schedules and night time working hours. But also some more specific to pilots, for instance travel across multiple time zones leading to jet-lag, and nights in hotels possibly effecting sleep quality.[16]

Several scales for assessing and discussing fatigue-level exists. One being the SP scale, measuring sleepiness on a scale from 1 to 7, and one being the KSS scale, from 1 to 9. In an effort to standardize these measurements the Common Alertness Scale, CAS, were proposed by Boeing along with Jeppesen. This is a scale ranging from 0 to 10'000, 0 meaning maximum sleepiness and 10'000 maximum alertness.[18]

3.1.2 Fatigue Risk Management System

Fatigue Risk Management is applied in a so called FRMS. It is this system that allow airlines to move away from the regular FTL. The structure and components of the FRMS is defined by the airline applying it, they are also responsible to provide evidence that the system is secure. The role of regulatory organs is to oversee the system and approve or disapprove it. They also continuously sample the performance of the system to make sure it gives proper level of safety. [17]

This type of safety work brings both pros and cons, in being more flexible. Given that different airlines can implement their own system, the measures to increase safety can be evaluated and improved upon iteratively. This also gives the opportunity to airlines to formulate standards that that are closer linked to the every day activities of the particular airline's pilots, instead of having one fixed standard applied across the industry.

The type of FRMS Proposed by Jeppesen includes several components in securing the flights. A central part is the Boeing Alertness Model, or BAM. BAM is a scientific model that predicts the pilots alertness level. The model can be fine tuned by replacing the default values for some parameters (like HSL and Diurnal type) to further increase the accuracy.

Along with the alertness prediction, many other components are integrated in the FRMS. One of them being the CrewAlert application, in that it can help the airline to collect fatigue data and incident reports from their crew members.[19] One other component in the FRMS(also using BAM) is the scheduling system, in pushing the schedules to reduce fatigue in the crew. Other components includes the scientific community, in sharing data and improving related models, as well as follow-up evaluations and overall communication of the importance of fatigue awareness.

3.2 Goal-Directed Design

Goal-Directed Design (GDD) is a design method which puts the user's goals in focus instead of their tasks. Given the goals the design team can find a more optimal solution for the problem. GDD consists of six phases, as seen in Figure 3.1.[20]

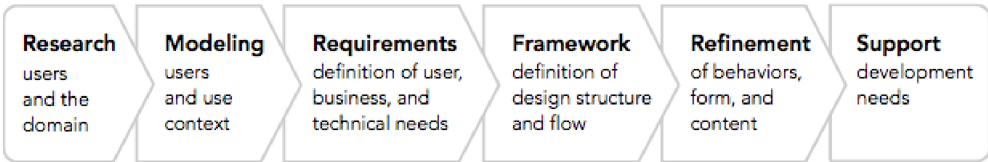


Figure 3.1: The six phases of Goal-Directed Design

3.2.1 Research

The first phase of GDD is to research the domain and its users. This is done by using ethnographic field study techniques (observations and contextual interviews) in order to gather qualitative data about current and/or potential users of the product. This data is then compiled via behavior variables into behaviour patterns and goals.

3.2.2 Modelling

The behavior patterns are used to create personas which reflects potential users/user-groups. A persona describes how a specific user, representing a user group, interacts with the world in there everyday life and what their goals and aspirations are. With the help of this description a tailored design can me made. Note that personas also can be used to model the customers (as opposed to the users) of a product, this to strengthen the eventual sales pitch of the product.

Usually highly refined personas are made, taking a month or more to create for most design firms. But Friess indicate that they are not used enough in decision making to validate the amount of resources spent on them. Furthermore, Friess points to research indicating that to many personas are created when there are usually only up to three personas being used.[21]

3.2.3 Requirements Definition

From the personas, context scenarios of the product is produced. A context scenario tells a story of how a persona uses the product in some context. From the context scenarios the functionality- and requirements specification of the product can be produced, which technology and data is needed for the product to match the context scenarios.

3.2.4 Design Framework

Here, the first sketches of the product is made using for instance paper prototyping of the interface and the general flow of the product. When should the user be presented with what so that it fits their mental model? As work progress, higher quality sketches are created.

3.2.5 Design Refinement

When a rough design has been made and approved, higher fidelity prototypes of it are produced. Here the details of the interface are created and honed until a complete design has been made.

3.2.6 Design Support

In case parameters around the project changes, such as time restriction, the design team should be ready to change design to fit the new time frame. This also includes working the development team so that the chosen design can be developed in a reasonable time.

3.3 iOS Design

The iOS Human Interface Guidelines is a set of documents describing the standard UI elements for iOS and how they should be used by the developer in order to create an application that is intuitive for iOS users.[22]

But more than the general iOS paradigm has to be taken into account, for making a universal application the design on the different devices supported must work together. The types of devices to support include iPhone 4S and earlier, iPhone 5, iPod touch, iPad and iPad mini.

When designing for iOS, making a *Universal application* means that the application supports all Apple devices running iOS.

3.3.1 Design paradigm

Apple have tried, and to a large extent succeeded, in having most of the applications for iOS to look and feel alike. The input controls and icons are the same between applications which allows the user to only learn one control scheme that fits all iOS applications. In order to succeed with this they have created a lot of free content for iOS developers as well as writing the iOS Human Interface Guidelines, telling developers how they should use them in order to develop a successful application for iOS.

This also goes hand in hand with another point the guidelines push on quite hard; that one should focus on the main task of the application. The user should also always feel that he or she knows where they are in the application's hierarchy and knows where they came from and are going.

3.3.2 Cross platform iOS design

The Human Interface Guidelines gives a number of tips and directions for iPad specific design, specifically for going from an iPhone design. They direct one to focus on increasing the interaction opportunities, given the extra visual and gestural space of the screen. This is recommended as opposed to broadening the functionality just because you can, given the extra screen real-estate.

A specific comment regarding the iPhone 4S and earlier compared to iPhone 5 design, regarding the extra space: "*Don't use the extra space to display an additional bar or banner.*" [23]

Focus should instead lie on expanding the area concerning the main task of the window. For the iPad this is further expanded on; to keep the hierarchy shallow and to reduce full screen transition by using pop-ups for lesser tasks.

3.4 Cross Platform Design Framework

Some guidance in cross platform design is given by the iOS Design Guidelines, but to increase the cross platform compatibility another framework was studied. This framework was proposed by Wäljas et al. and is used for analyzing and developing cross platform interfaces.[24] It builds on a study conducted on web interfaces for services on mobile phones and desktop computers. To enable simultaneous iPhone and iPad design this framework could be used, albeit with decreased benefit given the partially changed context. The framework targets user experience and bases itself on three components as follows. *Composition*, regarding functionality, *Continuity*, the ease of moving from one platform to another and *Consistency*, how consistent the interfaces of the different devices are.

The framework and related study did not say anything specific about touch based smartphone designs transfered to tablet designs, but rather smart phone to web interface, but it still provides an interesting way of thinking about the cross platform aspects of design.

3.5 Information Visualization

It is clear that the visuals of a mobile application have to take into account the possible circumstances the application might run under. This was explored by Pombinho, as he proposes in his mobile information visualization *Computation Contexts*; one must think about the following four perspectives of the context of the application. *User context*, what the user is like in personality and what goals the user might have (according to GDD), *Physical context*, the physical surrounding environmental constraints, *Temporal context*, when the application is used and *Historical context*, what the user did last time etc. These contexts combined, claims Pombinho, give an wide array of characteristics to analyses within the application.[14]

As Shneiderman explores multi-layered interfaces, which builds on dividing interface components according to the scheme of novice, intermediate and expert user segments, as described by Cooper et. al.[20] This generates both ease of use for the different user groups, but the overhead and possible confusion of having multiple interfaces is not to be neglected. The multi-layer interface design method applies more thoroughly to desktop and laptop computers, whose applications usually are not as narrow in feature-sets as mobile applications, but the method still might give clues on how to divide information objects and interface components.[25]

The representation of data require different levels of cognition for its reception. On the lowest level data is said to have sensory immediacy. This means that we do not have to process the visual data to see the information, we just “feel” it. Examples of this include orientation of objects, grouping of objects in different ways, etc. Colors, in particular, is an effective way to convey data. Being aware of these points and playing on their strengths is beneficial to any design.[26]

3.6 Related applications

In surveying related applications no corresponding applications were found. However, a large amount of flight logging applications exist for mobile devices, the most prominent being Logten. It is described in short below, along with RosterBuster, since it is tightly coupled to CrewAlert by its schedule export capabilities.

RosterBuster

The RosterBuster application builds on the web-service with the same name, allowing the user to upload their Roster in order to access it in a easy to read format on an iOS or Android device or as an online calendar. The RosterBuster web-service also provide importing of schedules into CrewAlert. The application is available on both the Apple App Store and Google Play. RosterBuster is show in image 3.2.[6]

Figure 3.2: The RosterBuster application, here representing a schedule with map view of the trips.

Logten

The Logten application by Coradine, is a flight log book for both iPhone, iPad and Mac computers. It lets the user track flights and flight hours in a comprehensive way. It also lets the user track how long he have been on duty and how long he can continue working without breaking some specified FTLs. This application is relates to CrewAlert in that it also keeps records of flights. The difference is that its primary use is to enter past flights. The user can however enter future flights as well, giving a verdict on the legality of the flight patter according to the set FTLs. This differs from CrewAlert in that the FTL are what CrewAlert is helping to replace. The application is shown in figure 3.3.[27]

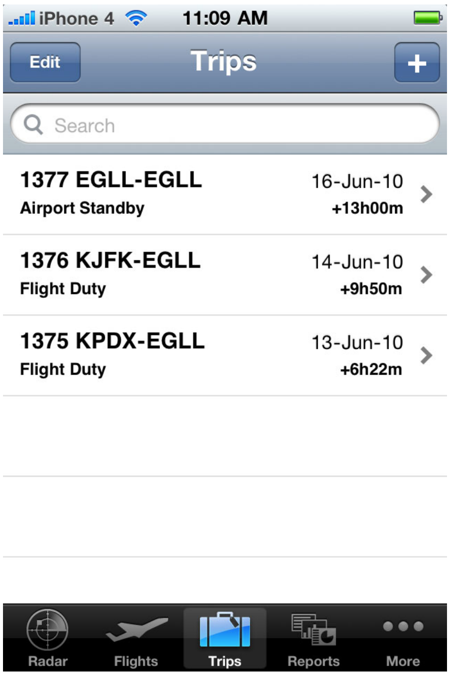


Figure 3.3: Logten by Coradine here showing a view of taken trips.

4

Methodology

This thesis followed the Goal-Directed Design method as described in section 3.2 with some alterations. These alterations were made to make the development more *agile* in order to fit the short time frame of the project. The method was performed with the development framework Scrum with two week sprints.[28]

Scrum were used more or less during the entire work period, with some adoption to non-development tasks. A Post-it wall were used as scrum board to ensure physical presence of the priorities and organization, leading to proper levels of engagement with the tasks.

Parallel to the scrum meetings, meetings were also held with the supervisor at Jeppesen, along with a team of people involved in FRM. In complement to this semi-regular meetings was also held with our supervisor from the Interaction Design and Technologies master program.

Implementation was performed in the iOS Integrated Development Environment (IDE) Xcode. With Mercurial for version control.

Around this base more design methods were used as described below.

4.1 Interviews

Interviews were used to gather information about both Jeppesen and the CrewAlert users. The interviews were held in a loose discussion formats to get a flow of conversation going. This resulted in more data about the interviewee's general ideas and thoughts, rather than strict answers to closed questions. The interviews were held in a office environment and to help both the attitude, making the interviewee more open and positive to the experience, and the blood-sugar of the subject, coffee and cookies were provided. A friendly and relaxed tone were used to get the interview smoothly along. Another method to make the situation more pleasant is, as a interviewer, not sitting directly opposed to the interview subject, or seeking exaggerated eye contact. This methodology on interviews was provided by Lantz.[29]

Each interview were preceded by a prestudy and a brainstorming session, to explore the space of possible topics. This resulted in mind-maps which were translated into interview questions.

4.2 Brainstorming

As a base design for the brainstorming sessions the Human Centered Design (HCD) Toolkit by IDEO were used.[30] The aim for each session were for it to be open, topic centered and creative. The goal of the brainstorming was both to generate as many good ideas as possible, but also to generate bad ones, to help get them off our minds. Criticism on others' ideas were done in a positive way as to inspire new ideas.

The tools used were pen and paper, whiteboard, post-its and computers. On the computers sites like Google image search were used to generate input to new design concepts. The sessions lasted roughly 40 minutes each, until the idea flow started to slow down. After each session the best ideas were discussed further and documented.

4.3 Parallel design

In complement to the brainstorming this activity of separate design helps avoiding homogenization of ideas. First the boundaries of the design goal at hand was discussed, and when these were established each participant spend roughly 10 minutes formulating a solution. After that each participant explained his idea to the others, then a discussion followed, finding good and bad sides with the solutions. Then the process was repeated until a good solution emerged. This method was used to generate whole solutions where many component ideas were integrated. This meant that at the explaining stage, there was many pieces that could be borrowed and expanded upon. After a few iterations solid ideas usually emerged, and the designs started to stagnate.

The parallel design method used during this project was not anchored in the literature, similar methods are however described by other authors.

4.4 Personas

Personas are archetypes of the intended users. They are created by studying the users, or other reference material, and written as character with some personal information, life situation and personality. Along with information relating to their positions and their tasks. The point of a persona is to give the corresponding user group a name and make it easier to design for them. The main pitfall when creating personas is the risk of falling back on stereotypes. One of the ways of creating and using personas, the one used here, is described by Cooper et al.[20]

4.5 Prototyping

For prototyping, four different mediums were used. First of all were paper sketching, which was used for early concept design. By using A5-papers, real sized iPad mock-ups could easily be done. To enable collaborative sketching, also enabling larger scale, whiteboards were used. They could be iterated upon far more easily and it also allowed for collaboration in a grater extent than paper.

Later in the process both Xcode and Photoshop were used to prototype. Xcode for its Interface Builder, making it very fluent to directly manipulate interface elements, and Photoshop for its ease of use in creating high definition designs and look and feel concepts.

4.5.1 Withboard along with iPad camera

During the whiteboard sketching, the use of an iPad and its camera made it possible to get a hands on feel for the design in as it progressed. On the whiteboard a large frame was drawn, mimicking the iPad screen. With the help of a rough estimate on the number of iPads the frame was in height and width, the proportions of the frame was made quite precise.

The frame was then populated with a design suggestions and iterated upon. An iPad was used continuously to take pictures of the white board. The photos could then be viewed directly on the iPad, and be evaluated in the right context with proper scale of every component drawn on the whiteboard.

With this method one gets the benefits of the large area of the whiteboard, enabling sketches of higher fidelity, while still remaining in touch with how the design will look on the actual device.

5

Process

5.1 Prestudy and initial analysis

To initiate the work we studied the ACM Digital Library, in order to get the current best practices for mobile platforms development.[31] The articles relevant to this project were however few and far between. Some interesting frameworks were found that could help, but the main theoretic background came from the iOS Human Interface Guidelines.

The development of FRM is in its early stages, which meant that no corresponding application existed for us to look at. But to get a feel for the type of application that we were about to develop we surveyed applications related to CrewAlert. One important application is RosterBuster, the application that enables automatic decoding of schedules for import into CrewAlert. Another application is Logten, by Coradine. This application is the leading digital logbook for pilots. More information on these applications can be found in section 3.

To get a further understanding of the work at hand, we studied the domain. This started with the Jeppesen portfolio of products. Both to get a frame of reference to what the product owner company does outside the area of this project, but above all to get a firm idea of the need the developed application is intended to fill. Furthermore this elicitation could help point out the path in witch to further improve the application. The master thesis that spawned the application in the first place were also read as reference.

Much information was found on the company's website.[5] After combing the company's website for information on their general product portfolio, and the parts relating to FRM in particular, it was time to get more in depth information. As the GDD-method suggests it was decided that interviews with subject matter experts would be held. To help the gathering of domain knowledge some interview techniques and survey literature were read to increase effectiveness and take better use of the potential.[29, 32]

Two interviews were held with the project's responsible and product owner, Tomas Klemets as a subject matter expert, on two different subjects. First, on Jeppesen's product portfolio in general and the parts more closely related to the work at hand. This information lead to a better understanding regarding the larger scheduling systems, safety work and longterm planing of crew and, of course, FRM. The second interview was held about the target audience of the application, pilots. This interview resulted in second-hand information about the every day life of a pilot, their educational needs, common attitudes of the industry, but also how the application would fit into their routines.

Here, a general discussion emerged resulting in information that could be applied during the design process. See appendix C for more information on the interviews (in Swedish).

Initially there were indication from the company that a groups of pilots would be included in the design process, with a series of meetings. This was, as it turned out, not the case. This resulted in that no real user study could take place before the actual development.

Prior to our work on CrewAlert, a usability survey had been constructed, that was to be sent out to current users of the application. The main focus on the survey was usability and a the recently added *Fatigue Mitigation* functionality. The survey was distributed to the focus group of users that had been recruited for this testing. Since this survey already had a usability focus, and it was deemed important to keep the length of the survey down, our addition was limited to one question. The responses of the survey were collected through an on-line form.

Brainstorming sessions were used to find ways the application would in fact help the customer and, in turn, the company. This generated word clouds and mind maps on the subject and ideas on what direction to take the application. This step also opened the chicken or egg discussion. Whether the application should do what the users wanted of it or if it should create the need itself, in trying to become a driving factor. Since the application concept is quite novel, and not just being a way of replacing some older way of working directly, this discourse were taken a long way, and was held open during the next few steps of the process. This also spoke for asking high level questions.

In acting within the Goal Directed Design methodology the question was to be kept on a high level. The uncertainties about in which way to take the concept mentioned earlier also pointed towards getting more information that would be relevant in high level concept design. There was, however, a risk that the question would be too vague, and not yield useful answers. Especially since the users had tested the application and had a very firm idea of what it is today. These points formed the question: *"What role would you like CrewAlert to take in your everyday life?"*

In total this question got twelve answers, of the total 16 that answered the survey. Answers came in and were read and taken into consideration as the prestudy went on. This, along with stakeholder discussion lead to the construction of two key focus points as explained in section 5.1.1.

During the development of CrewAlert, RosterBuster released a new version of their mobile software including features such as a map view, showing upcoming trips and weather for upcoming layovers. This contributed to the down-prioritization of similar features from our concepts.

5.1.1 Result of analysis

From the analysis two key focus points were derived. These were:

> *How to improve the interface and interactions in order to accommodate for new users?*
>
> *How to improve the longevity of the application, providing substance for the users after the novelty wears off?*

The benefit of the application is greatest when used regularly and for longer stretches of time. Both because the data is more conclusive but also because it has potential upsides for the airlines employing CrewAlert users. The first of the points is here to make people continue using the application once the novelty wears off. The analysis pointed out that the application was to some extent used just to *"play around"* with, and that the fatigue prediction was not being all that helpful. One survey response for instance compared it to a heart rate monitor (as used while exercising), in that it was awfully helpful and fun to use in the beginning. But when you learn the functionality, and on your own can estimate your heart rate, it quickly looses its fun.

This leads to the aim of bringing up the every day use of the application, broadening the focus of the application from just the fatigue risk perspective. This have to be done whilst keeping the benefits of the FRM in the (hopefully rare) cases a flight will be dangerously fatiguing.

The second point, to lower the learning curve, came up both from our own experience of having to learn the application and from the survey. This can also be seen from the large tutorial and e-learning material needed to get past the threshold of everyday use.

With these points we thought that the longevity of the user base could be increased. This then increases the value of the application from the perspective of the users, Jeppesen and the scientific community. From here on, these two points will be in focus.

5.2 Design activities

The work then continued according to GDD by creating personas which were to be used as a design tool. The initial step of persona creation were to specify behavior variables within the target group of the application.

A analysis of the application were also conducted, which lead to the documenting of the Functional and Data elements (as defined by GDD) present in the application. The elements gathered made it substantially easier to get a complete overview of the application and how to use it.

As a result of the study circumstances the application would be used under, eg. in the cockpit, were also identified.

5.2.1 Persona creation

Since the pilot profession is quite narrow, our focus group could not be used as hoped, no initial interviews were able to take place. This made searching for information from other sources a prominent task. Information were however found across the Internet in a variety of places. This information stemmed from non scientific sources like testimonial bloggs and articles.[1] And general information about their everyday tasks.[2]

After the online research and SME interview, three target groups were selected to base persona creation on.

The groups were:

(a) Novice young, and low in their career, pilots
This group were selected to represent the early technology adaptors.

(b) Intermediate, regular airline pilots
This group were selected to represent the grassroot base needed for the application to grow.

(c) Experienced pilots
This group were selected for their experience; by using the applicatin they could input data and help improove the application once it is released.

The type of personas created during this design project is provisional personas, due to their lack of actual personal observations.

5.2.2 Context scenarios

Upon the provisional personas context scenarios were created to help specify the functionality of the application. The scenarios describe activities one could perform with the application in a certain context. The scenario should describe the perfect product interaction, without specifying how it should look or work beneath the hood, as Cooper et. al. describes it; "*Think about the interface as being magic*".[20]

[1] http://jetcareers.com/becoming-an-airline-pilot.html by Shem Malmquist
[2] http://science.howstuffworks.com/transport/flight/modern/pilot.htm by Joel Freeman

The context scenarios include activities such as preparing mentally for a flight and workday, using the application in the context of the cockpit. How the sleep logging should be done on a regular basis. How the pilot could use the application to help him during flight delays.

The context scenarios were then worked upon and specified with brainstorming and parallel design. Mostly this was done within our group, but our supervisor at Jeppesen also contributed throughout the process in less formal brainstorms. The bi-weekly meetings with the Jeppesen stakeholders also gave input to the process.

5.2.3 Pilot interview

After this we managed to get in contact with a pilot from SAS[33], and conduct an interview with him. He was a middle aged man and fit the target group of our intermediate persona. The result from the interview were compiled and compared to previously gathered knowledge. The interview confirmed most of the insights obtained from the prestudy, but some were on the other hand contradicted. For instance the need for mental preparation, and planning ahead of time was toned down. This lead to a change in concept direction more towards making the application focus on providing information relevant in the moment. Another possible feature of the application, that of loggin, was also toned down. The need for personal flight loggs was not as big as we had found from other sources.

The questions were spawned by brainstorming based upon the previously conducted interviews. The question mind map are shown in image 5.1 (as made in Swedish).

Figure 5.1: Mind map which was translated into interview questions for the pilot interview.

The use of only one person as a reference for the target group is quite insecure, but the information provided about his line of work was still thought to be of great value. All in all, some re-thinking of the way in which to take the application were made.

5.3 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 CrewAlert as a planning and preparation tool. Before beginning a trip the pilot would use CrewAlert to visualize the trip and mentally prepare for it. Thus incorporating CrewAlert into the pilots workday.

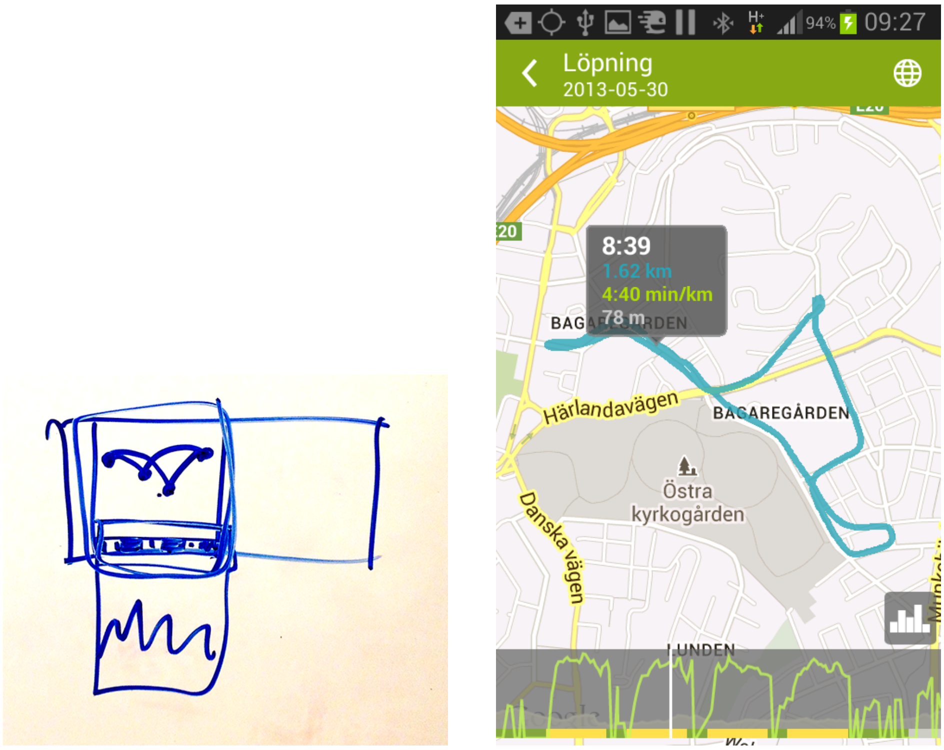
5.3.1 Development of early concepts

To do this a shift in focus was to be made in the application, becoming more of a planning tool and focusing on duties and trips and with the current focus, alertness, being a side feature. The center of the application was now to be a map view visualizing the flight paths of the duties for the upcoming trip on a world map combined with a time axis. And if the user dug into the trip the current graph view showing alertness would be visible, but the focus of the application was now to give the user something more practical and have the technical alertness prediction more hidden, used to influence the user without having to take up the entire screen.

To evolve this concept, inspiration were taken from an Android application called Endomondo, a GPS sports tracking application similar to applications like Runkeeper. The idea was that you switch up and down between the map view and the graph view. the schedule would be visible in both views, on the top in graph view and on the bottom in Trip View. See figure 5.2

This concept was however turned down, due to a number of reasons. One being the fact that the user needs to enter a complete schedule for this to work as intended, instead of just entering the first departure and last arrival of the day, and not the entire schedule. Another being that the map view would not depict correct flight paths, due to the special constraints of flight planing (as well as the great circle). After consulting with the product the owner at Jeppesen it was decided to be too big to implement fully and not falling within the range of what they saw the application to be. Although the concept as a whole were not to see the light of day, it still influenced us and we kept the main ideas of it going forward.

The previously discussed concept had been to large and not entirely in the right direction but the main ideas behind it where still valid. Therefore the same basic ideas was to be realized in another way. As a second concept that was spawned was an idea centered around information cards, displaying different pieces of information relevant for the pilot, in a stack like fashion. This idea takes inspiration from the the *Google Now* application for Android.[34] This application presents the user with context aware information cards, for instance weather information for your current location, upcoming calendar events and tram departures for the closest tram station. In Google Now the information cards make up a long scrollable list, the more relevant a card is thought to be, the higher up it is



(a) (b)

Figure 5.2: The Trip View design idea compared to the Endomondo map view. The left image shows the two way scroll of the trip view. Left/right to scroll in time, and up/down to switch between the trip visualization and the graph. The Endomondo map view shown was used as inspiration for the trip view.

in the list. The application can be seen in figure 5.3, displaying two examples of these cards.

These concepts were sketched both on paper and on whiteboard. The whiteboard sketches were made in a large scale, with proper height to width ratio. This enabled us to take pictures of them with the iPad, giving us a proper feel of the size and relations between objects, due to being in their right context; the iPad. This also lead to better documentation of the sketches, which was needed because of the volatility of the whiteboard.

This concept were iterated on until the end of the design part of the development process. The concept as a suggested iPad landscape look, shown in figure 5.4, were shown to the Jeppesen stakeholders. Much of what were presented were well received but some features, like the combined timeline and data system where down-prioritized as not being important to implement in the early iterations. Instead the focus was to be put on the cards and making the application universal.

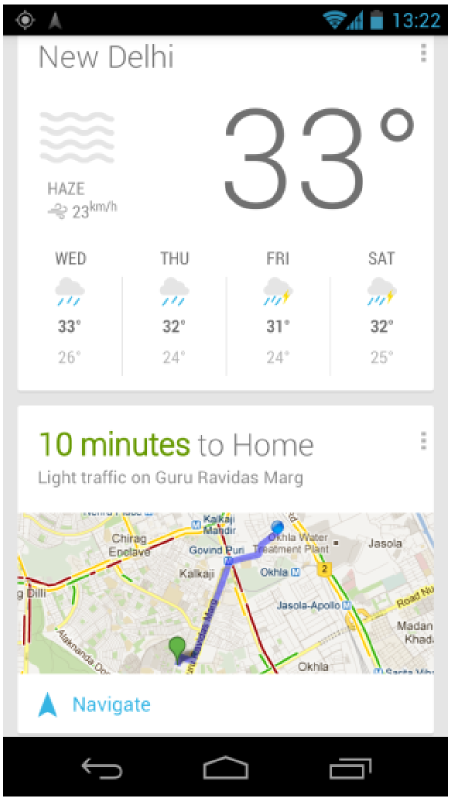


Figure 5.3: The Google Now application, used as inspiration for early concepts, showing two information cards.

Some focus was still put towards the information related to trips, but with information regarding the overall structure of the trips and less about how to mentally prepare for them. With information such as when the next trip was, how much spare time where to be had between flights, what was the weather going to be during this spare time, and so on. This concept were sketched as see figure 5.5. After discussions with the stakeholders the concept was locked to be centered around these cards, which were to be called panels.

5.3.2 Development of panel concept

The initial idea for the information panels was to have them focus on information around trips and duties, keeping with the focus of the new CrewAlert being more of a planning tool. As a natural part, alertness and FRM were brought in, both as a dedicated alertness panel and as part of other cards. Thus we combined the ideas and shifted focus back to FRM whilst still holding on to a few of the original concept ideas. This concept was helped by the pilot interview performed which was performed during this part of the development process.

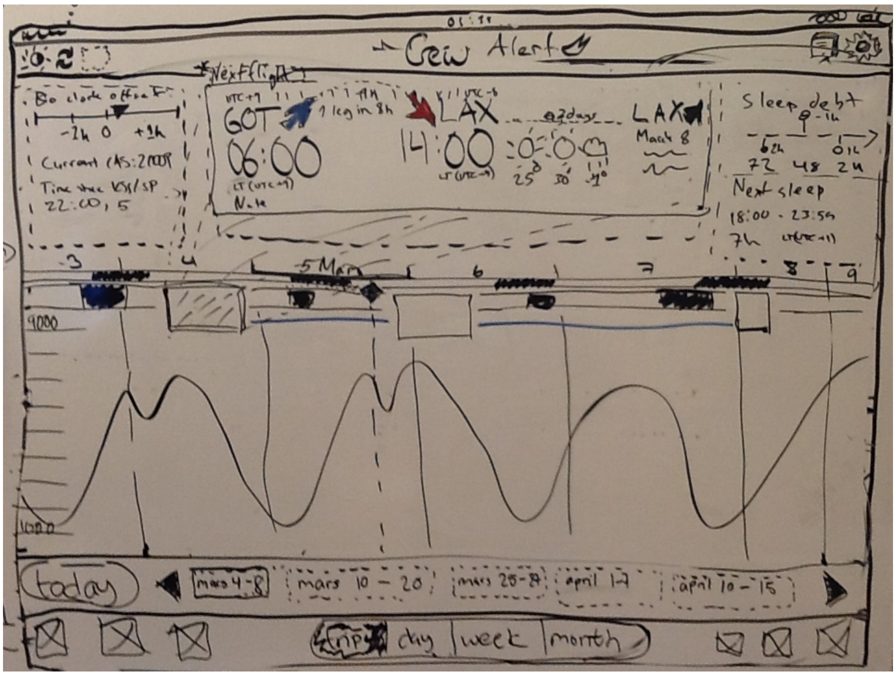
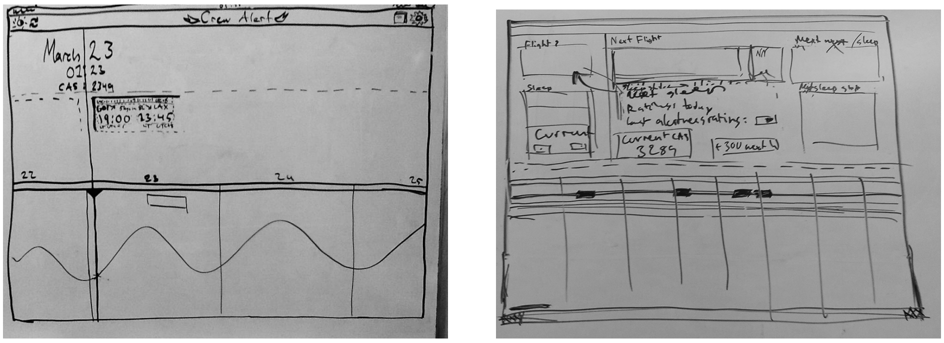


Figure 5.4: Sketch of second iteration on the information cards concept, as it was presented after our main design sprint.



(a) (b)

Figure 5.5: Aleternate design ideas for the information cards concept.

On the iPhone we created a new tab which were to be the new initial window of the application. This tab was to hold the information cards. After some mockups of this screen, one example shown in figure 5.6, the concept came to be inspired by the iOS notification center. This was done to benefit from the conceptual similarities between the two screens; which also give an overview of the current situation; showing time, weather, stocks etc., and provide information about upcoming events; showing calendar info.

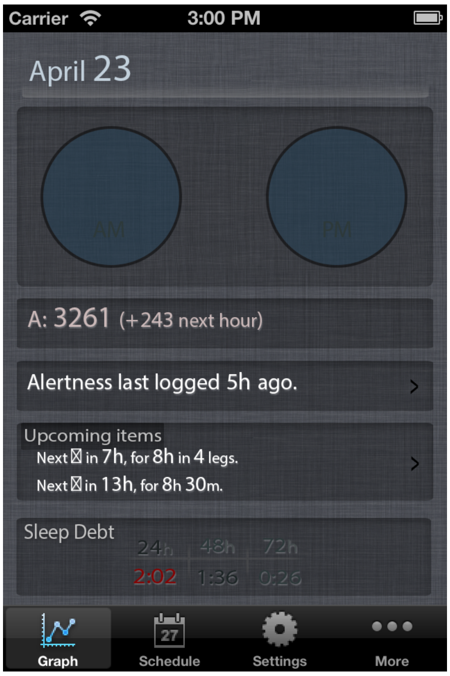


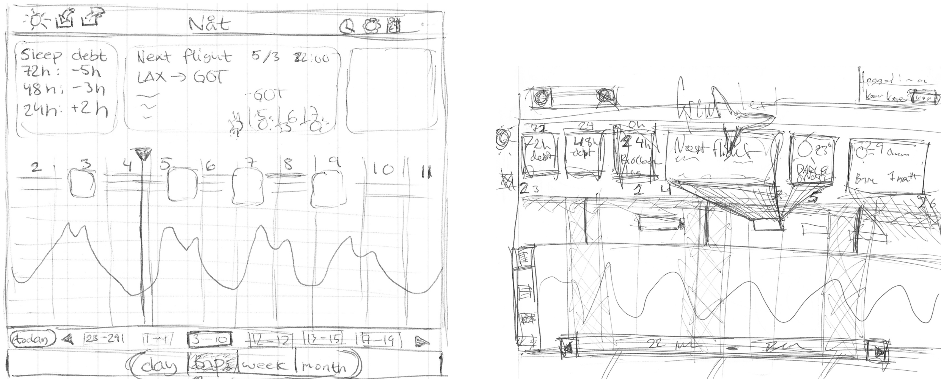
Figure 5.6: Early design of the now tab for iPhone.

On the iPad we split the main screen into two. Having the old graph view as one part and the new information cards as the other. We started by designing the iPad look, since it would be more challenging to design, due to its size, since we would not tale down the information density too much. The iPhone panels would easily fit in a new tab, and the layout did not need to be reworked to have both landscape and portrait as was necessary on the iPad due to the difference in user pattern.[22]

The iPad design work spawned many ideas of layout and interaction between the panels and the graph view. In figure 5.7(a) and figure 5.7(b) the two most influential ideas can be seen. These designs were then moved over to the iPhone and the notification center look, iterated on, and transfered back. This concept was the one iterated upon the rest of the project. The design work shifted between the iPhone and iPad devices, since they needed to be consistent between platforms. The switching of platforms could be seen as a design tool, that helped us think about the design in different ways when moving it back and forth between the platforms.

5.3.3 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



(a) (b)

Figure 5.7: Design ideas for the iPad look

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 sleepbar left the lightbar “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 lightbar 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 inputing 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.

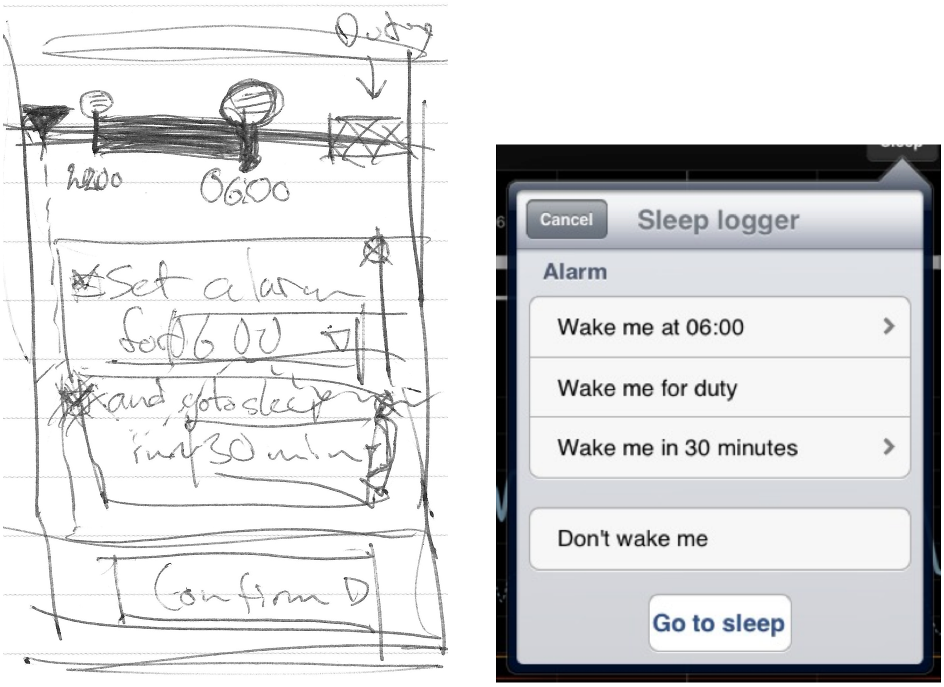
Alarm clock In order to help the user log their sleep, an alarm clock system was designed. This would allow the users to do their sleep logging implicitly while using the alarm clock feature. The alarm clock work similarly to sleep cycle-recognizing alarms, where you activate it when you go to sleep, 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 CrewAlert could make it more time effective to use. Since the user might want to check other information in CrewAlert 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 have 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. Some prototypes can be seen in figure 5.8.



(a) (b)

Figure 5.8: Prototypes of the alarm clock.

Local time concept

To make it easier to input time zones a *Local Time* option, referred to as LT, was 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.

5.3.4 Trip Bar

This bar has its foundation in the concept of CrewAlert as a planning tool. It was inspired by the bottom bar in Apple's calendar application for the iPad, see figure 5.10. The bar was at creation given the ability to show days, trips and months. The Jeppesen application Terminal Charts also gave some inspiration towards having a bottom bar, see figure 5.11.

The Trip Bar, seen at the bottom of image 5.4, evolved from the idea of a scrollbar that visualized the flights and fatigue, like a miniature version of the *Graph*. See image 5.9 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.

It was however found during the analysis that the application that these more regular stretches of time were not as interesting as the so called trips. This lead to the adaption of the trip concept from the lines connecting duties in the schedule, seen in figure 2.4(b).

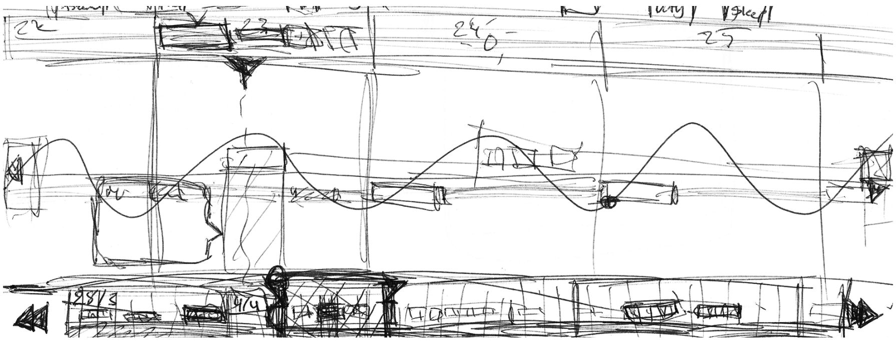


Figure 5.9: Easly concept drawing for the scrollbar visualization.

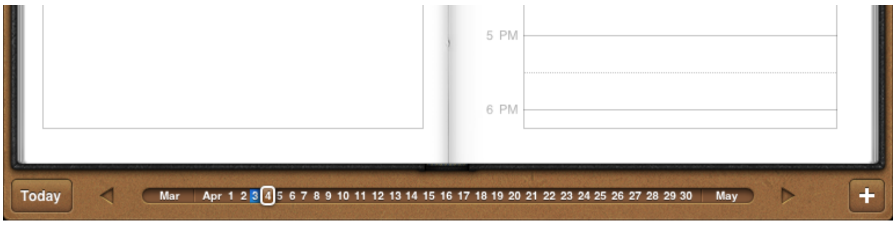


Figure 5.10: The date picker bar on the bottom of iCal for the iPad.

5.4 Panel design

The information panels work as a tool for user to prepare and plan flights. Therefore the main panel was the *Upcoming Duty* panel which contained information on and around the next upcoming flight.

Figure 5.11: The Chart picker bar on the bottom of the Jeppesen application Terminal Charts.

For a number of panels warning limits were used to alert the user if the values displayed were starting to get urgent for the user. The values are of different types for the different panels, but the coloring used was consistent across the panels. The application did initially already have two warning limits for alertness, in the *Graph* and *Schedule* view. These colors were also used for the new indications, to keep the application consistent.

5.4.1 Upcoming Duty

The Upcoming Duty panel started out as a next flight panel. Here information about the next upcoming flight was displayed. This information included departure and arrival airports and times, the warning level of the flight, a note (if one is entered by the user), number legs in the flight, time zone shift, time to spare between this flight and the one after it, the weather during this time, the departure time and airport of the flight after. A sketch of this can be seen in figure 5.12

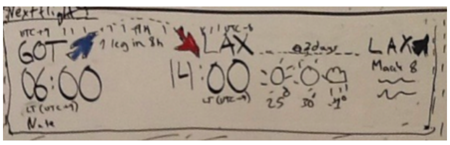


Figure 5.12: Whiteboard sketch of the next flight panel on iPad in landscape

The thought behind it was to first show the basic information about the flight such as times and airports but then focus on the planning ahead. This was done by showing how long it would be between the next flight and the one after that to indicate how much free time the user would have at the arrival location and also the weather for that time. This enables the user to easily decide upon cloths to bring, and what to plan.

Due to implementation time, and the fact that external services would have been needed, the weather section of the panel was dropped. The panels evolved through iterations, giving them quite distinct looks on the iPhone compared to the iPad. Second to space, the main reason for this is the context of use. The iPhone is more likely to be used in-between flights since it is always accessible in the users pocked, compared to the iPad which is more likely used either in the cockpit or at home. This calls for a here-and-now focused design for the iPhone.

Figure 5.13: An early version of the Next Duty card, with a suggested look-and-feel.

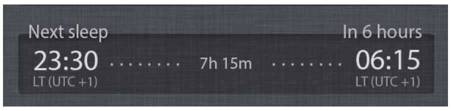
5.4.2 Next Sleep

The idea behind this panel was to give the user an easy way to plan their next sleep. They could know when they should go to sleep in order to keep up with the schedule in CrewAlert. The benefit might not be all that large when sleeping regularly, since we are used to always calculate what time to go to bed, essentially without thinking, if we need it. But the as part of the FRM perspective this panel give valuable information in terms of reminding the user of the importance of good sleeping habits. A great benefit of this panel is the showing of the unusual sleeping patterns, when the user for instance goes from day shifts to night shifts. In these situations the intuitive or natural sleeping patterns might not be the best ones.

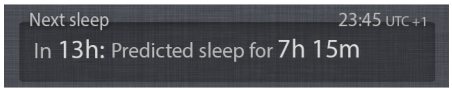
Initially this panel did show start time end time and the amount of alertness gained during the sleep (delta CAS). Some scenarios were played out and it was shown that the delta CAS points was not a good measurement, after noon naps do for instance often result in no alertness net gain, but rather a delay of the fatigue. The sleep inertia also affects this measure in a negative way. The idea to measure the alertness at the point where the sleep ends, and compare this to the alertness of that time if no sleep were to take place. This was also rejected, due to the lack of comparative value between this value and the actual CAS-values.

The panel evolved to not showing this information at all. The time format was also reworked to be relative, giving it less of a hortatory feeling. This also made more sense from the point of view that you at any given time should get the most valuable information from the Now tab. If the user for instance check in on the application a couple of times per day, the relative information will be more interesting since you get a updated value each time.

The look on the panel evolved from the notification center-look towards a more general pane-look, see figure 5.14. Here, the difference between relative and fixed times are also shown.



(a)



(b)

Figure 5.14: Design ideas for the *Next Sleep* panel.

5.4.3 Sleep Debt

This panel helps the user to keep track of his sleep debt which helps them become more aware of the effect of their sleep in a longer perspective. This makes the long term perspective of sleep deprivation more prominent. The sleep debt can be verified easily, and is not as “magic” as the CAS score, and the user can get a feeling for how the debt affects him or her specifically.

The sleep debt is the amount of hours of sleep the users is deviating from his set HSL and is shown in three different accounts, 24, 48 and 72 hours. The numbers showing the debt gets highlighted once pass a certain limit, to indicate sleep patterns that are either unhealthy or unsafe. The highlighting are done in two stages, first in amber and then in red.

Some iterations were made on the logic behind the warning limits, while the visual look have been kept the same since late concept development.

5.4.4 Alertness

From the beginning the alertness was a important piece of information to bring into the panels. Initially it was, as a initial non-design, displayed as its points on CAS. We came to realize that the CAS points, but also the graph, might look a bit sterile or theoretic to the new users. This, along with discussions with Jeppesen, lead to representation with a more graphic orientation. To give the alertness more of a interesting position, but also increase the focus of it.

So, to increase the pathos of the alertness measurement a CAS visualization was introduced. The initial idea was to make it a head, with a tank meter visualizing the CAS points and the back of the cranium being the indicator clock face, with level of alertness being represented as amount of gasoline.

This was however changed, to more accurately describe the effect of fatigue, to a battery. The battery as a metaphor have two benefits; it is not instantly fill up, as a gas tank, but have to be charged (as one usually does with the device at night). The battery is not an ON/OFF energy source like gas. The battery performs with lower and lower effect much like the way the body and brain works with increasing fatigue.

The indicator, the battery charge level, was given five levels one representing the red warning level, one representing the amber level and the remaining three were green. The range of the different states will be given from the CAS level at wakeup being (approximately) fully charged, and working their way down to the user set limit for amber.

Alternatively the red level could be visualized with an empty battery, with a red high light, and the amber level having only one bar. Next to this visualization there is also an image and text telling how long before a level will be gained or dropped. The gain or drop-time were later replaced with a clock instead, see next section.

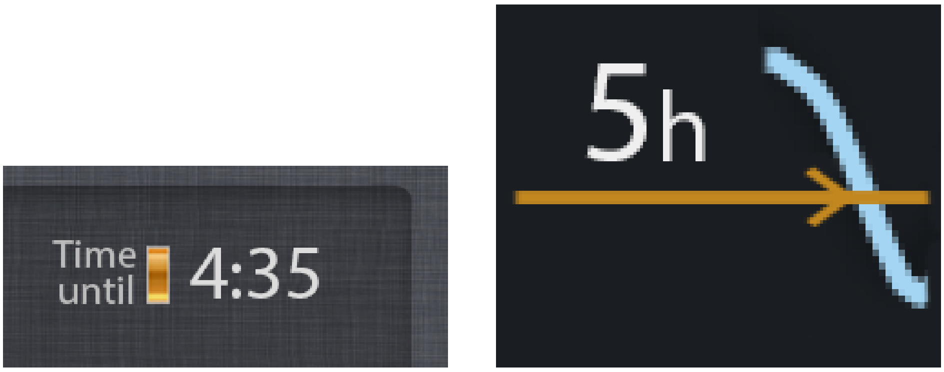
Due to the complexity of the amber and red limits each having a designated number of bars, along with the more imprecise feel of the bars, this was changed to a design where the battery level is continuous rather than discrete. Similar to the iOS charge indicator. This also lead to a complementary way of showing warning levels, namely a stroke outlining the head, also changing colors corresponding to the amber and orange warnings.

5.4.5 Clocks

The clocks were a natural part of the Now tab, they displayed the time in two different ways. One were a simple custom timebase clock. The additional feature of which was the local time feature. Depending on the user settings on their device it could be advantageous to have the current local or home time given. For example if the user has set their device to not update its clock when switching time zones it could be of interest for the user to see the current local time and vice versa. The other one was body time clock, showing the biological clock as predicted by BAM.

The clocks originated as a analogue clocks at the top of the new *Now* tab, both to give the screen a visual distinction and to not keep the entire screen filled with the same type panels. But when the silhouette head were introduced, they were naturally sized down to avoid bloating of the panel, and they were also changed to digital ones to further increase the look of the panel.

A third clock was later introduced to show the time that can be spent awake, until the alertness level drops below the next warning level. This clock went through a number of looks, two are shown in figure 5.15(a) and 5.15(b).



(a) (b)

Figure 5.15: Design ideas for the *Time to Warning* clock, showing the time left until the user reached the next alertness warning limit.

5.4.6 Self assessments

The way for the user to perform self assessments when we started was to either double tap a check mark icon in the *Graph* tab or on startup (were the collection screen would automatically open). Neither of these ways was pleasing to us since the check mark icon was unpliant and the log on startup could be annoying and also forced the user to log *directly*, instead of when they were feeling ready for it. The recommendations from the CrewAlert documentation is to take five minutes to calm down before making a alertness log entry, to avoid getting a measure of the alertness in the "heat of the moment".

To improve this we created a new button from which assessments could be logged. The label of this button would, if enough time had passed since the last assessment, be highlighted to indicate that the user should do a new assessment. The time is defined by the *Prompt interval* user setting, and can easily be changed, the feature can also be disabled if the user does not wish to log alertness. This would help the user to perform frequent assessments without forcing it upon them.

Since the check mark icon also took care of sleep logging, in certain cases, the label was later changed to *Log data* which was dynamically changed to *Data logged* after doing a data entry.

5.4.7 Acclimatization

One of the later introduced panels were the acclimatization panel. The point of this panel is to provide advice to users that wish to acclimatize in preparation of a trip. An early sketch of the panel can be seen in figure 5.16.

One point of evolution with this panel is the option to select what time zone to acclimatize to. This problem concerns both interaction and reliability of the prediction of the model, too much information can not be extrapolated and presented to the user as reliable. This led the solution to only let the user select a cardinal direction to acclimatize towards.

One potential design was to show advice for both directions at the same time, and the users choose what advice, if any, to follow. This resulting in more of a information panel than actual advice, but this could also help the user in his everyday use, reminding him of the effect of light exposure even when no acclimatization is wanted. This idea was however left out because, plainly it was too much information with too little value. On the contrary a button was chosen to enable switching. To not give arbitrary advice, an OFF option was also also introduced.

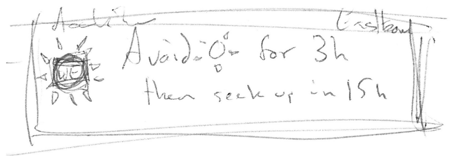


Figure 5.16: Early sketch of the acclimatization panel.

5.5 Further design

In addition to making novel concept designs and taking the purpose of the application in a new way, some pinpointed design efforts were also needed to increase the usability of some of the reused features of the application. In this section, the design process of some of these features will be discussed.

5.5.1 Scaling and orientations

Along with continued work on the information panels the universal aspect was discussed and designed. An issue was the much larger screen on the iPad. The *Graph* tab with the schedule and graph was not able to simply be scaled up to the entire screen without loosing information density in a negative way. This lead to the decision to scale the Graph partly and add new features to the remaining space, balancing benefit and minimalism.

Another problem area was the extra vertical space on iPhone 5 in portrait and iPad in portrait. For the iPad the use of space was not as pressing an issue, in part because the portrait orientation is secondary to the landscape orientation, so new features to fill the space were not really a preferable solution. This lead to the decision to keep the design as it were, with only fine tuning positions, sizes and ratios of the panels.

The iPhone 5 was put through more iterations of design suggestions. Among others to add rows of buttons at the bottom, add an information panel from the now tab and to introduce a miniature visualization as scrollbar, similar to the early Trip Bar concept.

This would however leave the iPhone 4S and earlier models left out. To avoid the danger of using different interaction patterns in the different phone models, a more simplistic solution was chosen. The solution decided upon was to simply scale up the whole view to match the larger screen of the iPhone 5. This was grounded in the iOS Guidelines in many ways, it states specifically that the extra space on iPhone 5 not should hold any extra buttons, and they also state that you should try and give more room to the main feature of the application.[22] The added landscape opportunity on the iPhones also made it less necessary to keep the information density up.

In figure 5.17 and 5.18 two high fidelity Photoshop mockups are showing the working progress of the iPad main view and the iPhone Now tab.

Figure 5.17: Photoshop mockup of the panels on iPad in the landscape orientation.

5.5.2 Clarification of interactions

One example was the *Create Scenario* screen. To create a scenario you need to give it a name, which was done through an alert box. The text field in the alert box is not all that visible and if no scenario name is provided and the *OK*-button is pressed, the dialog box will disappear, just to reappear as it was with no further feedback given to the user. This feature was changed to make it obvious that a scenario was created when the corresponding button was pressed. That a scenario needed a name was also emphasized. This was done by placing an input field in the table of scenarios, the user was thereby giving a name to an already created scenario.

Figure 5.18: Photoshop sketch of the now tab on iPhone.

The settings were also reworked for better understanding. This was done by grouping specific settings into sub-screens and moving items between the More and the Settings-tab.

One aspect of our two main design goals was to create clear interactions. The application did originally had some hidden interactions, interactions that the user had no reasonable chance of knowing about except reading of the tutorial. For instance, in the Graph view, the action to add duties and sleep/wake logs by double-tapping on the schedule.

Whether the user taps on the upper half or on the lower half of the schedule decides what add-screen is shown. Also, the add-duty screen is shown when the user double-taps on predicted sleep. This might be confusing to the user, if he wants to edit the predicted sleep.

As a solution to this problem buttons were introduced. These buttons were placed on the the side of the schedule part of the Graph view on the iPhone. This was above the check mark icon, making them a row of three buttons equal in size and over-all look, with the addition that the check mark were turned into a button rather than a icon, making it take only one tap instead of two to open the *Data Collection* screen. On the iPad they were placed in a tool bar on the edge of the Graph view, giving them roughly the same position.

5.5.3 Importing and exporting of data

Another feature that were thought to be needed was the ability to sync data over the cloud. The original meaning of this feature was to support use over multiple devices for a single user, given the extension of the application to support both iPhone and iPad. This feature would also enable the user to bring his data with him when upgrading device, but also for backup, since the amount of personal data stored and the effort put forth by the user to log this data is quite substantial.

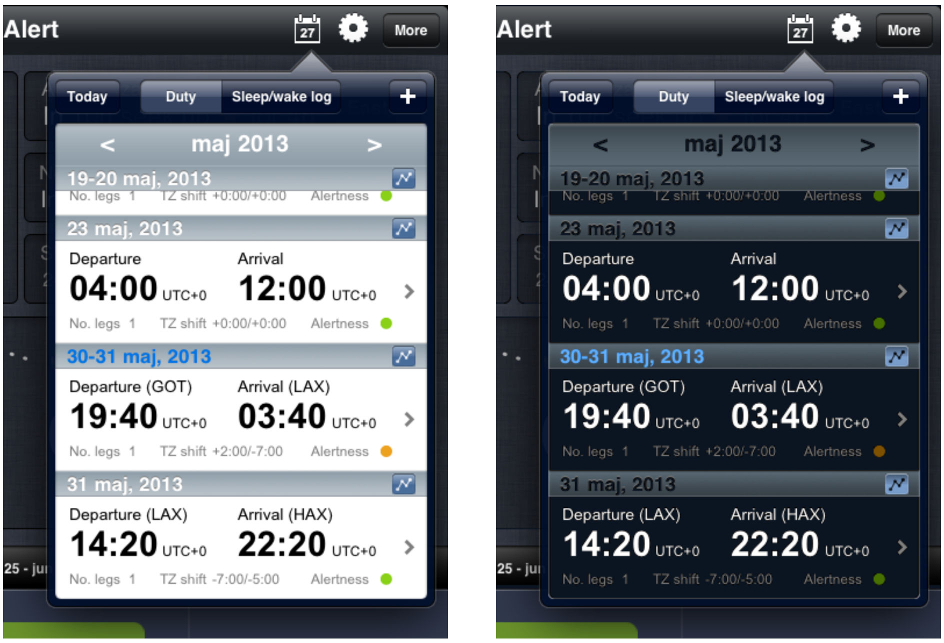
The iPad version will be used, to a larger extent, in cockpits, making the possibility to sync data across devices extra important. This point however dismantled one solution, that of using iCloud as a base for this feature, since it would result in both privacy and policy problems in multi-user iPads.[35]

To import data from another service, such as Jawbone Up, would also be an option, given that they have extra functionality such as sleep quality and more accurate data.[36] This would be a interesting opportunity to bring in more detailed data into the system, but it would require much work getting to the more interesting parts of the data and especially to incorporating them into the model.

The development of these features is a standing wish and essential for the future of CrewAlert. However, there was not enough implementation time during this project to focus on these features.

5.5.4 Dimmer and night mode

To satisfy the constraints of use in the cockpit, for the application to be used while flying, a dimmer was deemed necessary. The feature is also present in other Jeppesen iPad applications that are part of the Electronic Flight Bag. This made the design as easy as copying the one from the preexisting applications. To further increase the night time in-flight usability, of the application, a night mode would be preferable. This was however not designed more specifically than the main concept of removing all bright surfaces and replace them with darker ones. This feature was not implemented, but an ad-hoc test with a simple inversion of brightness can be seen in figure 5.19.



(a) The schedule in regular mode. (b) The schedule in night mode.

Figure 5.19

6

Result

At the end of the development a range of design solutions and interventions were formulated, some fully implemented, some left as unimplemented design suggestions. Apart from the new solutions, CrewAlert's base functionality was also upgraded it to support iPad and iPhone 5, making it *universal*, as well as supporting all orientations.

The solutions are here sorted as a subsection of either of our two main goals; *To facilitate recurring use* or *To smoothen the learning curve*. Overall, these key points made the application focus shift towards a *here-and-now* perspective. The details resulting from each goal sums up as follows.

6.1 Facilitating recurring use

In order to motivate the user to come back to application and continue the use over a longer period of time we provide new and relevant information for them. The idea for this was information cards providing information on what was going on in the moment, always information relevant at the time the user picked up the device and used the application. This was realized with panels containing information about upcoming events, alertness and more.

On the iPad the panels are placed directly in the main view, which is split into two parts, the panels and the graph. On the iPhone on the other hand they were given their own tab, the *Now* tab. The icon for the now tab is a heart-beat shaped waved.

6.1.1 Upcoming events

From the user's schedule and predicted sleep pattern, information is extracted and presented in a relevant format as upcoming events. These events include the next duty, the duty after the next one, and the next predicted sleep. The standard way in which time is presented in is relative, i.e. how many hours or minutes it is left until a certain event.

Next duty panel

The most important panel among the upcoming events is the next duty panel, see figure 6.1. This panel shows the user's next inputted activity, either a flight, a set of flights (as a duty) or some other work-shift, for instance a *dead-head* flight or *ground duty*. It shows when the duty starts both in time-left and as actual time and how long it is, the departure and arrival airports and for flights (and dead-head flights) the number of legs is shown. There is also room for a note, if the user chooses to enter one.

The alertness on takeoff and at top-of-decent for the duty is shown as visualized by arrival and departure airplane icons. These icons change color depending on the alertness at the time of each of these events (see figure 6.2). The same limits are used as those that determine the color of the duty in the schedule of the *Graph* view. Next to the airplane icons the airports are shown, as their three letter IATA code.

This panel exists in four different versions. It differs between the iPad and iPhone formats with each format having two versions, one for next duty, and one for the duty after the next one. The main difference between the iPad and iPhone formats is the amount of information displayed, since the iPhone screen real-estate is substantially smaller, even as the panels got an exclusive tab.

(a) iPad version.

(b) iPhone version.

Figure 6.1: The next flight panel, with (b) also showing a user note.

The iPhone version uses the relative time until the duty starts as the main time indicator while the iPad version uses the fixed time. Both panels present the information both ways, but which one was emphasized differed between them. The iPhone version however does not show the end time for the flight to save space. The iPad version also displays the time zone shift between the two airports.

The panel for the duty after the next is similar to that of the the iPhone, with the only difference being the relative time. It shows the duration between the end of the previous duty to the next instead of the actual time left. This goes for the iPad version as well, here only the takeoff part of the card is visible, see figure 6.2.

(a) iPad version. (b) iPhone version.

Figure 6.2: The duty after next panels, showing low alertness at takeoff.

Next Sleep panel

This panel, shown in figure 6.3, provides information about your next predicted (or pre-logged) sleep. The time is given in a relative way; the time left until start of sleep. The actual time, along with its time base, is also presented. The time left also uses two warning levels, one for when the sleep is getting close, and one for when the user starts to miss out on a substantial amount of sleep. The time gets colored amber and red when reaching the first and second warning level respectively. The first warning levels are set to one hour prior to the sleep and the second is set two hours after the sleep was predicted to started. The panel switches sleep five hours into the sleep, to account for early awakenings without showing strange information.

(a) iPad version. (b) iPhone version.

Figure 6.3: The *Next Sleep* panel.

6.1.2 Alertness information

Several pieces of information are provided to indicate the current alertness. These include CAS score, a *Time to Warning*-clock, a *Body Time Clock* and a panel showing the current sleep debt. These are presented below.

Alertness visualization

The alertness panel's main feature is a visualization of the current predicted alertness. The visualization is a silhouette head containing a battery representing the alertness, rising from the cerebellum as a staple. The battery is a charge level indicator, like on the iPhone charge screen, where full battery equals high alertness. The remaining staples of the battery are colored green to indicate high alertness, and as it drops, it will first turn amber and finally red. The alertness threshold levels correspond to the users personal settings. As the battery staple gets smaller and smaller as its value drops the area showing the warning also decreases in size. To help the problem that the most severe level of warning is the least visible, a stroke was added that also changes color to indicate the low alertness. The stroke also makes it stand out more from the background. See figure 6.4.

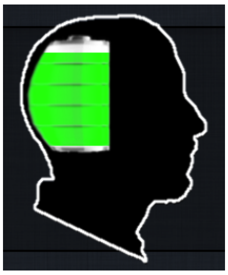


Figure 6.4: The silhouette head showing the current alertness.

Time awake to next warning level

This panel, see figure 6.5, shows how many hours and minutes you can stay awake until your predicted alertness reaches its next warning level. When the displayed time is less than five hours, the clock changes color to amber. Once the amber warning limit is reached the panel changes its visualization, showing red text instead.

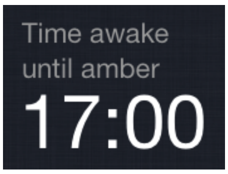


Figure 6.5: The time awake to next warning level panel.

Log Data button

This button, see figure **??**, is used to log data. It indicates with color coding how well the user is keeping up with the logging. If no sleep or no self assessment is logged the last 24 hours it will be highlighted.

The information shown works as a Data Collection button to make it easy to update the alertness score, once the user notices the alerting label. While in survey mode the label and button will be highlighted once you exceed the time interval set as *Prompt interval*, with witch the user can set to auto-open the *Data Collection Screen* on application startup, to remind himself to take the self assessments.

(a) Button when data has not been logged the last 24h.

(b) Button when data has been logged.

Figure 6.6: The Log Data button.

Sleep Debt panel

This panel, see figure 6.7, shows three calculations of the users sleep debt to indicate sleep patterns that are unhealthy or unsafe. The sleep debt is shown in three different *accounts*; for 24, 48 and 72 hours. This calculation is based on the user's *Habitual sleep length* setting. This value is then compared to the number of hours the user have sleep during this period. It does not matter if the sleep is logged by the user or predicted by the model. The sleep debt also has the two warning levels, set at 60 and 75 percent of the habitual sleep length. These limits also correspond to amber and red warning highlights.

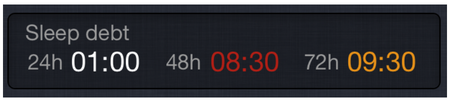


Figure 6.7: The *Sleep Debt* panel showing substantial sleep deprivation.

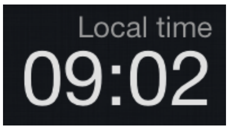
6.1.3 Additional information

To utilize the complex calculations made by BAM and give the user more substance in the application, some additional data is extracted from the model and presented to the user in the following two ways.

Clocks

The Alertness panel contains two clocks, see figure 6.8. The first clock is a regular clock, with the possibility to set time zone setting independent from the other settings in the application, or to local time, making it follow you as you travel.

The second clock is a so called *Body Time Clock*, showing your predicted biological clock, as calculated by BAM.



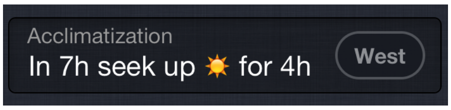
(a) The regular clock

(b) The Body Time Clock

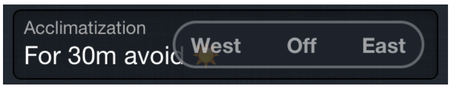
Figure 6.8: The two clocks in the Alertness panel

Acclimatization panel

This panel, shown in figure 6.9, tells the user how to seek up or avoid light exposure in order to acclimatize in a certain cardinal direction. The panel contains its own setting button, where the user can select what cardinal direction to acclimatize in. The panel also have an *OFF* option, disabling the panel. This panel also uses times relative to now to indicate when to avoid and seek up light.



(a) Acclimatization panel giving westbound advice.



(b) Acclimatization panel with the settings button open.

Figure 6.9: The acclimatization panel.

6.1.4 Dimmer

A dimmer that lowers the brightness of the device screen so that pilots can keep their night vision whilst doing night flights.

6.2 Lowering the learning curve

The second part of our result is the overall design updates to help the usability and lowering of the threshold for learning the application.

6.2.1 Simplified schedule in the Graph view

The two sleep bars (model predicted and user inputed sleep) in the schedule part of the Graph view were merged, as can be seen in figure 6.10. This was done by superimposing one on the other, with the user inputed data overriding model data. To indicate where the data comes from the users, the data has a black border around it. Predicted sleep still have their light-gray color whilst logged sleep has a pale blue color. The light bar was moved to be on top and now also applies a dark or light layer to the entire graph depending on night and day. The line and arrow indicating the current time is also changed. It now stretches all the way across the schedule, and the anchoring of it is placed on top of the light bar.

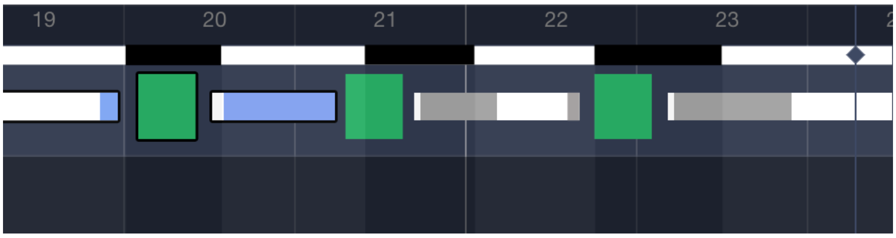


Figure 6.10: The simplified schedule in the Graph view, also showing the now-indicator on the far right side.

6.2.2 Simplified input of user data

The previous system indicated to the user that data that should be entered by displaying a data collection screen at startup. Instead we propose that this should be done by smart highlighting and the enabling of easier data entry.

Highlighting of missed data

In the graph view, if a predicted period of sleep lies in the past it is seen as missed logging of sleep. It is then highlighted by a red border as in figure 6.11. This highlight fades over time so that missed entries that are more then three days into the past are no longer highlighted. The *Log Data* button in the *Now* tab is also highlighted, but only for the last missed sleep, keeping up with the "*here-and-now*" perspective of the screen.

Duties who's surrounding sleep is confirmed will also have the black border.

Easier entry of data

In the graph view the user can double-tap on white and gray areas in the schedule and spawn a data entry screen. Here the user chooses what data to enter through a

Figure 6.11: Graph view schedule with missed sleep entries, highlighted with a red border.

segmented controller, see figure 6.12. The user then enters the time period of the entry along with entry-type specific information. The preselected entry type will be the last used type, with the exception of highlighted predicted sleeps which will preselect the *Sleep* entry type. Start and stop time of the predicted sleep is also pre-entered in this case, based on the times for the predicted sleep period the user tapped. The old way of entering data through the Data Collection Screen is still possible.

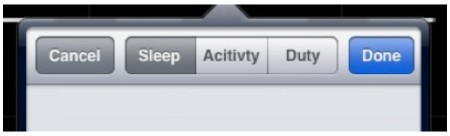


Figure 6.12: Entry of data in the Graph view.

6.2.3 Graph view navigation tool

On iPad an extra navigation tool was was added to help the overview of the schedule and to complement the *Graph* view. This bar, known as the Trip bar, shows the upcoming trips in a compact format. The content of the bar scrolls which enables the user to get a quick look at the next months schedule, as well as being able to quickly scroll to any trip by tapping it. The Trip bar also contains two (symmetrically placed) buttons, one for jumping to today, and one for selecting a date to jump to with a date picker.

This bar also shows transitions to new months and years, looking similar to trips but somewhat different in design, so the user easily sees the time perspective.

The Trip bar also is accompanied by a zoom view controller. With the zoom view controller enables the user to decide how long the left and right arrows will move the graph view. As well as what will be shown in the Trip bar; trips, days or months. See figure 6.13.

Figure 6.13: The Trip bar.

6.2.4 Scroll between time blocks

The navigating in the Graph view were extended to include a alternative to use the arrows in the segmented controller. As a alternative one can scroll beyond the edge of the Graph view, resulting in a effect similar to that of Twitter's *pull-to-refresh* gesture.[37] When the scrolling reaches a certain limit the window jumps to the next block of time. The scrolling distance is dynamically dependent on the width of the Graph view, being smaller on the iPhone and larger on the iPad. The jump between the windows is animated.

6.2.5 Easier time zone entry

During time zone entry, see figure 6.14, the user can can scroll the UTC offset to either *Home time (HT)* or *Local time (LT)* by tapping or spinning the left spinner to either *HT* or *LT*.

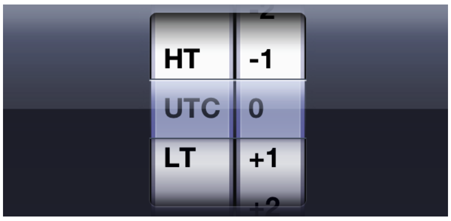


Figure 6.14: A time zone entry spinner.

6.3 Final look

In the figures 6.15 through 6.18 the final look of the application can be seen. Due to time limitations not all of the above design could be realized in implementation. The following features explained in the result was not implemented: *Graph view navigation tool*, *Easier entry of data*, *Highlighting of missed data* and *Simplified schedule in the graph view*.

Figure 6.15: Final look of the iPad version in the landscape orientation.

Figure 6.16: Final look of the iPad version in the portrait orientation.

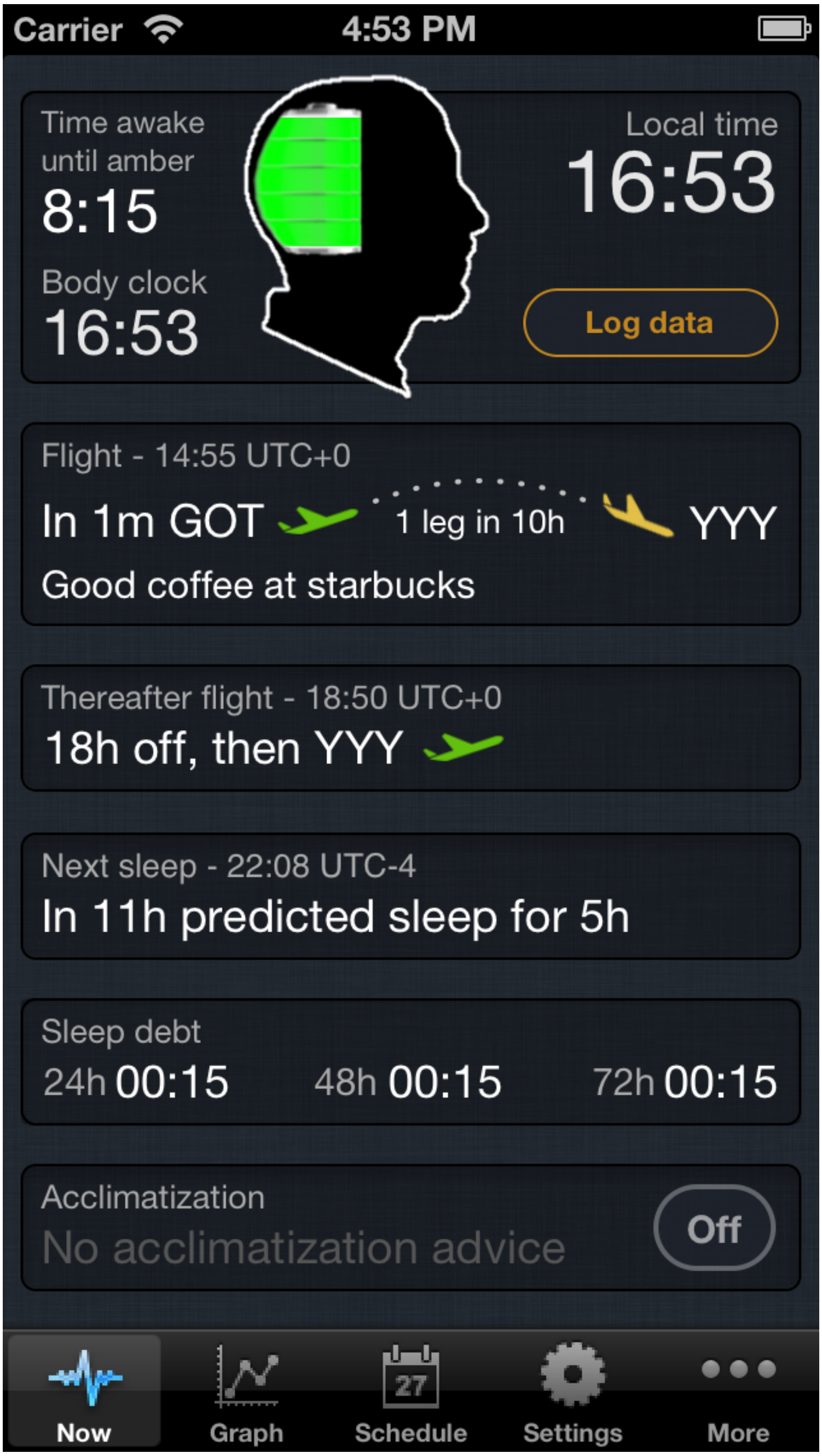


Figure 6.17: Final look of the now tab on iPhone 5.

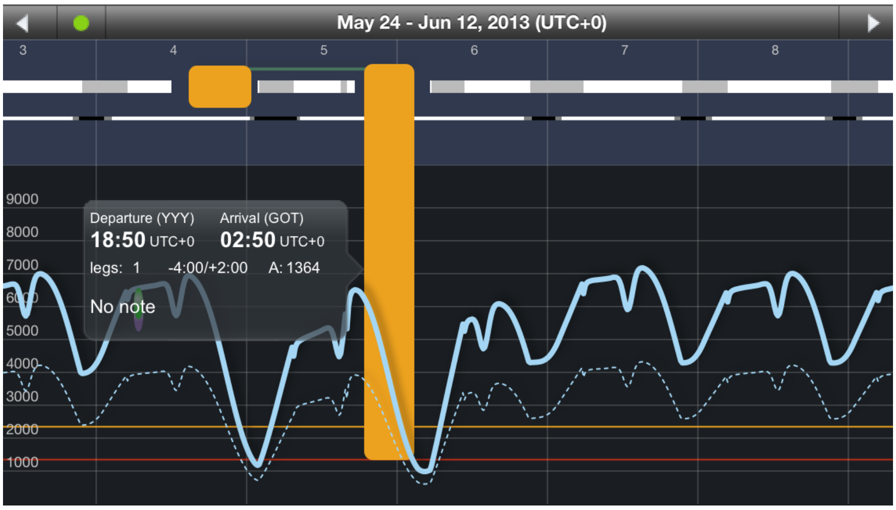


Figure 6.18: Final look of the graph view in the landscape orientation on iPhone 5.

7

Design Analysis

The over-all design concept we choose for CrewAlert was to find ways to bring the user back to the application and keep using it over longer periods of time. In order to add this we wanted to present updated and relevant information to the user every run. Our solution for this was to present the data already available in the application in a way that would help the user.

How the concept of the application would fit the target group is based on data gathering from semi-reliable sources and should not be seen as definite truths. Our assumption is that it is beneficial and interesting for a pilot to plan ahead and keep track of his professional schedule in order to bring out the most out of their time, as well as to keep updated.

Given that our initial analysis of the domain and users are correct, we feel that the application meet the needs fairly well. Especially given that the application was quite feature rich to begin with, not giving us enough time to redesign it from the bottom up while keeping all the features (which was a prerequisite by Jeppesen). The design of the *Now* tab should give an appropriate look-out of the current situation of the iPhone, while the combined *Now* and *Graph* tabs on the iPad gives an easy overview. The amount of interaction upgrades also made the application substantially easier to learn and use.

Along with the larger concept design, many small steps were taken to improve the individual features. Those with larger design relevance are analyzed below.

7.1 Timeline

The re-design of the time line, superimposing the logged sleep on top of the predicted sleep, was according to us an important step in the design. However, it can be argued that the changes to this visualization is simply a compression of information with little gain. When this is done, confusion could for instance occur about what sleep is logged and what is predicted. Their juxtaposition limits the sensory immediacy of the logged sleep, in that it does not stick out as much as it did in the original design.

The proposed solution did however counter this problem is two ways; they coloration and stroke. These are two of the more easily recognized ways of differentiating visual data, according to [26]. With the coloration uses the idiom of a *disabled* button, graying it out, it fit the pale blue look. The gray color gives the predicted sleep a feeling of unpliancy or *"disabledness"*. This making it comply with the fact that you can not modify it directly, since it is calculated by the application. The pale blue logged sleep on the other hand, begs to be interpreted more like an "activated" version of predicted sleep, in that it is confirmed, albeit with due modifications.

The fact that the sleep logs created by the *Fatigue Mitigation*-functionality also does have a red border works into this system. Because the sleep is not actual logged sleep, even though it is documented as a sleep log. This is analogous in meaning that also this sleep should be overwritten whit the actual sleep as time progresses.

The overlay of actual sleep on the predicted sleep makes the gap in sleep opportunity prior to a duty invisible if sleep was logged during this time. This problem is however not all that bad, since to sleep outside of the sleep opportunity is real possibility, and the fact that it looks strange in the visualization simply reflect that, given the current schedule, it actually is weird behavior from the user. The anomaly can for instance depend on things such as miss in correcting the schedule, it can then give the user a queue to update the schedule.

The highlighting of sleep that is missed is based on the memory of the user. Since the highlight works as a prompt to make the user enter sleep for that time, it would not be reasonable to highlight sleep so far back in the past the user could reasonably not recall their sleeping pattern for that time. The three day length of the period is however chosen by qualified guessing.

7.1.1 Light bar

The solution enables the user to see how both flights and CSFM-advice (see section 2.3.2) lie compared to night and day. It is also very easy to spot unusual daylight activity, after long flight in the east or west direction, since the entire graph now helps visualizing the shift in night and day. The downside of this solution is that the color toggle might be thought of just as a calendar day indication, making it look erroneous when the light pattern is skewed, albeit possible this is seen as a minor problem. The solution for the light bar also make the relation between night/day and the users circadian rhythm more prominent, which might bee interesting to the user.

7.1.2 Add to log

The changes made here would assimilate the Schedule tab and the *Add to log* screen, in putting the logging of sleep and duties in the same screen, increasing the coherency. It also allows for the sleep bars to be merged, since what of the two bars were tapped previously determined what type of screen to open.

7.2 Data collection

With the new *Log Data* button for accessing self assessments and and the highlighting of missed sleep logging, there was no need to keep the Data Collection Icon (see section 2.3.2). It could be seen as unnecessary to remove it, to be "on the safe side" and consider old users in having with multiple ways of accessing the Data Collection Screen. But since the interaction with the icon was quite arbitrary, it was removed and the *Log Data* button was left the only way to manually access the screen.

7.3 Panels

The context of use differs somewhat between the iPhone and the iPad. The iPhone is something the user brings with him or her close to all of the time, rather than the iPad witch is more specific in context of use. This shows in the panel design, with panels on the iPhone are more focused on giving information on the fly. On the iPad the times presented is focused on the actual hour, compared to the iPhone where focus lies on the relative times. The *Now* tab is also a overview tab which has the sole purpose of giving a quick overview, and with it being separate from the Graph view it could take on more of a current perspective, without the time-line spanning outwards, taking focus from the moment.

On the Next Duty panel; to have the arrival plane represent takeoff rather than top of climb alertness was due to actual risk present. Takeoff is a situation that in and of itself lowers fatigue, with the the added pressure on the pilot. This makes the top of climb a point in time where falling asleep might be a present danger. However, where the highest risk is in this context is not really for us to analyze.

7.4 Further Design

The time zone picking utilities were received with mixed feelings by the stakeholders. The issue was that potential confusion when the user selects HT or LT the selection does not stick, but instead rolls back to the UTC-option in the middle. It was kept the way it was with the argument that if the selection would stick, the two possible scenarios would, with +4 as HT for example, be to present the time zone as “HT +4” or “HT +0” both of witch can be confusing to the user. The alternative, to have the HT / LT picker rollback to UTC can of course be interpreted as an error, for instance that *Home Time* is not set, or that it can not be chosen. Especially in the case where the HT is already the selected time zone, since the hour-half of the spinner in this case would be stationary.

The compliance with the other Jeppesen applications was focused on color scheme and popover design. This gave corresponding color schemes, with the dark blue seen in the result section. The new look for the tutorial, and brightness control were also inspired by other Jeppesen applications.

Regarding popovers on iPad, it is good for the user to have similar views on the iPad compared to the iPhone, to help the transition between the two devices.

7.4.1 Icons

The sun icon given by iOS as the ASCII sun-character were chosen as acclimatization icon. This was chosen rather than the readily available icon for the *Fatigue Mitigation* advice icon for light exposure, which could be interpreted that the given advice had something to do with the *Fatigue Mitigation* functionality, which it not the case. The brightness control on the other hand were given the same icon, to comply with the other Jeppesen application as well as the color schema in the panel where it is placed.

The other in-app icon was the heart-beat shaped wave form, for the *Now* tab. This gives the feeling of acuity, with the panel content is giving an *at-a-glance* overview. The heart-beat shape could also reminds the user of the health aspects of sleep, which could make the application more accessible, since it might be seen as quite technical and sterile. Obviously, it also in some way resembles the alertness graph, which could be seen as a added novelty.

8

Discussion

We crated several concept ideas to solve the problems that we discovered with CrewAlert. Due to time limitations only a few of these were implemented and fully thought through. In the end we managed to extend the device and orientation support of CrewAlert from iPhone and portrait only, to supporting all devices and orientations. Along with this we added a new *window* through which the users could decrypt the information contained by the application. This *window* displayed both regular information in a new way, as fetched from the *Graph* and *Schedule*, and uses the model to generate new information for the data available. All this data is presented in an accessible and user friendly manner.

The information displayed may or may not be relevant to the user since this has not been tested on actual users but instead iterated upon in discussions with stakeholders.

We are in some ways happy about the result we produced. The result of the project can be seen as two halves. One being the design solutions and the other what actually got implemented. We are happy with both on an individual level, given the time we spent on them. We think that the design solutions proposed for the schedule and *Sleep/wake* logs in the *Graph* view would help a lot in lowering the learning curve. We are also pleased with the way the panels turned out, and the amount of implementation we managed to do in the short time frame we had. What we are not so happy with, is the fact that we walk away form this Master Thesis with two "incomplete" products, with a design not polished enough to implement straight away, and a implementation that is not fully polished to be a App Store release candidate. Work will continue on CrewAlert after this project, performing user tests and polish the current implementation so it is releasable on the AppStore.

8.1 Methodology

In hindsight we can safely say that given the time frame of the project and the intended end goal, Goal-Directed Design (GDD) did not work. GDD worked very well to create new design concepts and ideas of paths to take the application, but it was too slow. Since the intended goal was to create a new version of CrewAlert which was, if everything went well, to be released on the AppStore at the end of the project. GDD led us down a development path where we spent the greater half of the project purely on design and thinking followed by a short time period of crunch implementation.

To aim high with the design was not the best way to meet the actual requirements in the application development cycle. This could be seen as a problem with the GDD method, that design and realization are too separated. Not that one should mix designing and constructing all together, but a closer and more fluent connection between the two would be preferable. The reality check of bi-weekly meetings was a must for not making the goal unreachable. Although the initial interaction centered designs would have had a lot more to offer in usability and novelty in interaction.

The method used, was overall very good for concept development, but it was not so coherent with the style of development needed to efficiently fix the usability problems with such a broad application. This lead to somewhat of a growing apart between the goal concept and the actual implementation. The implementation work had to start before the design was complete, given the short time frame and the amount of usability problems found. So, had we not moved more towards agile development the implementation would not have stretched as far as it do now.

We do not blame GDD as a design method but instead our use of it. Going into the project we were due to, among other things, lack of experience, not completely synchronized with the project's stakeholders regarding what the end product was to be. We were focused on creating a whole new design concept for CrewAlert whilst Jeppesen wanted a new version of the application which supported iPad and included some new functionality, releasable to the App Store. This led to a split focus which ended up with us spending the first half of the project creating new concepts and ideas, most of which were down prioritized into limbo by Jeppesen due to time limitations. Followed by the second half were we moved away from the strict steps of GDD and instead started working according to the Extreme Programming (XP) method in order to faster implement the parts of the proposed concept that we were able to with the given constraints. The change was not made in discrete step but instead a gradual change over time as we realized that time were slipping away from us, whilst we still wanted to able to close the loop before the end. Along with this change we also stopped working according to the scrum framework.

We think that the right method to be used during this project would have been XP along with the scrum framework. If the project however would have had twice the length we think that the initial development method, GDD, would have worked best.

8.2 Future work

There is still much work to be done with CrewAlert, with many features designed but still waiting on the sideline, some for approval by stakeholders and some for implementation. Along with those the entire application would benefit from a general *"face lift"* since the current look and feel seems to us quite outdated and bland. This point is more acute now than before given the recent update of the iOS to version 7, bringing a totally new look.

It would be interesting to further study the design solutions proposed for the schedule and *Sleep/wake* logs, to see if they would bring the usability increase we think they will. Since they were not tested in a real world environment, and only by groups of developers, designers and stakeholders involved with the project. To do a double blind comparative study with real users, with one group testing the changes applied and one group testing the original version, would be very interesting.

Apart from implementing more of the unimplemented design ideas, it would be interesting to have a user test of what is implemented, up to this point. We would of course say the changes greatly increase the usability of the application, but without user studies, little can be said in terms of actual results. Did we enhance CrewAlert or were we just adding arbitrary novel interactions and unnecessary eye candy?

9

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 sum up all of this work in a Master Thesis turned out harder than expected given the broad field of work. Our interest started out in the usability and design parts, but functionality and usefulness turned out to take a lot of the time. 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.

The work on an application for both the iPhone and iPad was interesting, and gave a challenge in design that sticks out from regular design work. It could be seen both as a obstacle and a benefit to design in this way. The fact that you could move design choices between devices and iterate back and forth gave it the project interesting development style. It was also interesting work in a larger cooperation, albeit we could not benefit from the surrounding organization more than from those brought in by Jeppesen at the start of this project.

The end result was a complete concept along with additional design explorations for Jeppesen to take into account. The CrewAlert application will now continue on the trajectory we put it on. We are happy with the produced result and in the future lies validating the work with more user studies, as well as continued work on FRM in this mobile setting.

Bibliography

[1] Jeppesen, Fatigue risk management (Feb. 2013).
URL http://www.jeppesen.com/frm

[2] D. Hellerström, H. Eriksson, E. Romig, T. Klemets, The Best Rest, AeroSafety World Magazine.

[3] C. H. Österlund, J. Widlund, A mobile application for flight safety: The CrewAlert, Master's thesis, Chalmers University of Technology and University of Gotehnburg (2010).

[4] Boeing, About us, boeing in brief (Jun. 2013).
URL http://www.boeing.com/boeing/companyoffices/aboutus/brief.page

[5] Jeppesen, Jeppesen homepage (Jun. 2013).
URL http://www.jeppesen.com/

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

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

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

[9] M. Ziefle, U. Schroeder, J. Strenk, 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] A. Merritt, Replicating Hofstede: A study of pilots in eighteen countries, University of Texas., 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, MobileHCI '10, ACM, New York, NY, USA, 2010, pp. 493–494.
URL `http://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, MobileHCI '10, ACM, New York, NY, USA, 2010, pp. 495–496.
URL `http://doi.acm.org/10.1145/1851600.1851723`

[16] T. Akerstedt, R. Mollard, A. Samel, M. Simons, M. Spencer, in: Meeting to discuss the role of EU FTL legislation in reducing cumulative fatigue in civil aviation, ETSC, 2003.
URL `https://www.eurocockpit.be/sites/default/files/Akerstedt-Mollard-Samel-Simons-Spencer-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%20for%20Operators%201st%20Edition-%20English.pdf`

[18] Åke Olbert, D. Hellerström, T. Klemets, A Comprehensive Investigation of Flight and Duty Time Limitations and their Ability to Control Crew Fatigue, Jeppesen, 2011.

[19] Jeppesen, The boeing alertness model (Feb. 2013).
URL `http://ww1.jeppesen.com/documents/aviation/commercial/BAM_IV.pdf`

[20] A. Cooper, R. Reimann, D. Cronin, About Face 3: The Essentials of Interaction Design, Wiley, 2007.

[21] E. Friess, Personas and decision making in the design process: an ethnographic case study, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12, ACM, New York, NY, USA, 2012, pp. 1209–1218.
URL `http://doi.acm.org/10.1145/2207676.2208572`

[22] Apple Inc., iOS Human Interface Guidelines (Jan. 2013).
URL `http://developer.apple.com/library/ios/#documentation/userexperience/conceptual/mobilehig/Introduction/Introduction.html`

[23] Apple Inc., iOS Human Interface Guidelines - transition case studies (Jan. 2013).
URL `https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/TranslateApp/TranslateApp.html`

[24] M. Wäljas, K. Segerståhl, K. Väänänen-Vainio-Mattila, H. Oinas-Kukkonen, Cross-platform service user experience: a field study and an initial framework, in: Proceedings of the 12th international conference on Human computer interaction with mobile devices and services, MobileHCI '10, ACM, New York, NY, USA, 2010, pp. 219–228.
URL `http://doi.acm.org/10.1145/1851600.1851637`

[25] B. Shneiderman, Promoting universal usability with multi-layer interface design, in: Proceedings of the 2003 conference on Universal usability, CUU '03, ACM, New York, NY, USA, 2003, pp. 1–8.
URL `http://doi.acm.org/10.1145/957205.957206`

[26] C. Ware, Information Visualization: Perception for Design, 2nd Edition, Margan Kaufmann, 1999.

[27] Coradine, Logten (Jun. 2013).
URL `http://coradine.com/universal/`

[28] K. Schwaber, J. Sutherland, The Scrum Guide, Scrum.org, 2011.

[29] A. Lantz, Intervjumetodik, Studentlitteratur, 2007.

[30] IDEO, Human-centered design toolkit - second edition (2009).
URL `http://www.hcdconnect.org/toolkit/en/download`

[31] A. for Computing Machinery, Acm digital library (Feb. 2013).
URL `http://dl.acm.org`

[32] J. Trost, Enkätboken, Studentlitteratur, 2012.

[33] SAS, About us (Jun. 2013).
URL `http://www.sas.se/Om_oss/`

[34] Google, Google now (May 2013).
URL `http://www.google.com/landing/now`

[35] Apple, icloud (Jun. 2013).
URL https://www.icloud.com

[36] Jawbone, Jawbone up (Jun. 2013).
URL http://www.jawbone.com/up

[37] L. Brichter, User interface mechanics, US Patent, 2010, uS 8448084 B2.

A

Thesis proposal by Jeppesen

Flight Safety Thesis Work at Jeppesen

Prescriptive flight and duty time limitations for airline operators are gradually being replaced and complemented with a so called Fatigue Risk Management (FRM) approach. Jeppesen is the leading supplier for providing airlines around the world with functionality for Fatigue Risk Management.

This thesis aims to support the change now imposed on (or enabled for) airline operators by extending the functionality and usability of a Jeppesen iOS application called CrewAlert. The idea is to promote more extensive crew usage world-wide of CrewAlert, and thus the familiarization to fatigue predictions and fatigue mitigation functionality. This is done by making the app more useful in operation by using the larger iPad format now present in many cockpits but also adding new functionality.

The work includes:

- Survey pilots and interview Jeppesen personnel to identify pain points and opportunities.
- Prioritize new functionality and suitable usability improvements.
- Develop and conduct usability test with a reference group of airline pilots
- Deploy the updated app on the iTunes Appstore.

The technical platforms will be mainly Xcode for iOS development but also Python and Java deployed on Linux for server-side functionality.

We are looking for two self-sufficient (and highly motivated) individuals with some prior experience in iOS development, skills in interaction design/usability and interest in the aviation industry. Being fluent in English is a requirement. We hope to find two applicants with somewhat complementing skills: software development and interaction design.

The work will be conducted at our office in Gothenburg with an iterative development approach (typically two-week iterations) guided by a Jeppesen reference group assisting with both technical, scientific and business guidance. We are flexible on the starting date but would like to see work commence latest February 2013.

For more information about the Jeppesen FRM functionality, please visit www.jeppesen.com/frm.
Send your application to tomas.klemets@jeppesen.com

B

Thesis proposal

Master Thesis Proposal

Anton Persson and Johannes Andersson

January 23, 2013

Introduction

The proposed master thesis is to be performed along with Jeppesen as a part of their Fatigue Risk Management[1] work. Fatigue risk management, or FRM, is a new take on otherwise quite fastidious laws on the working hours of commercial airline pilots. With FRM Jeppesen helps consolidating and improving their flight schedules by taking scientifically predicted fatigue in to account. FRM is meant to help flight crew to manage the risks that comes along with long hours in a better way than saying you are too tired 07:38 and fully able-bodied at 07:39. FRM provides a way of upholding the safety whilst more realistically adhering to the reality of fatigue.

This proposal was made based on a request by Jeppesen, who had developed a FRM app, CrewAlert, as a previous masters thesis, and this proposal is in regard to replacing the previous app, giving the crew members the ability to use both the iPad and the iPhone, as well as make a more usable and streamlined experience for the user.

Stakeholders

Jeppesen is a company that focuses on systems for navigation management. They are based in the US with locations both in Gothenburg and across the world. In the Gothenburg offices the focus lies on scheduling of crew for air traffic.

For us, this is a interesting opportunity to prove our interaction design skills in real surroundings. This thesis also gives interesting possibilities to develop our skills in mobile development as well as networking and server environments.

Focus of our thesis

The focus of this thesis is to redesign the current CrewAlert application by Jeppesen. Apart from just redesigning the application, it will also be made universal, that is, making the same app run with customized views for both iPad and iPhone (with all screens).

Initially focus will lie on evaluating the pre-existing app and perform user tests with a focus group of pilots to see what the users want from an app like this. The focus group will also be utilized for testing along the development of the app. We will also analyze the app internally, as well as conduct case studies of iPhone to iPad conversions. The work does also contain designing and implementation of new features which will help the usability of the app.

[1] FRM, `www.jeppesen.com/frm`

An important part of the development will be to analyze the information visualization, along with the interactions. Challenge lies in presenting the information in a way that enables sound interactions, and works well in the specific conditions of the cockpit.

The development will be done using methods integrating both the end user and domain specialists, and other important stakeholders, in as large an extent as possible.

Research question

> *To redesign and extend the functionality of the CrewAlert application by Jeppesen in order to improve its usability and usefulness.*

Planned result

The planned result of the thesis will be the new CrewAlert application. The results could also contain a user study of the redesign to see if and in what ways it has improved the usability of the application. Along with this conclusions will be drawn on the development of a app in this specific setting, as well as pitfalls discovered either during our development, user testing or with the pre-existing app.

Work process

The development process for the thesis with be iterative, as suggested by Jeppesen as well as ourselves. Our goal is to accommodate the environment and select the working process most suited for our situation once deployed on site. We are preliminarily planning to work in a scrum like or agile fashion with two week iterations. This will be done with user integration, knowing the hardships of working close to the users.

Time plan

[W1-W4] Learning the domain
The first weeks will be spent learning the domain. This includes FRM, iOS development, current CrewAlert code, context domain and Jeppesens internal work flow.

[W5-W8] Analyzing CrewAlert
These week will spent conducting an analysis of CrewAlert's current design to find out what can be improved upon, what might be missing and what is redundant functionality.

[W10-W15] Design prototyping & testing
These weeks we will spent on creating design ideas and test them with the help of prototypes to see if they feel valid. Interaction with the end users will also occur here. At the end of this a finalized design should have been created. Design include the look and feel as well as functionality.

[W15-W16] Implementation & testing
These weeks will be spent implementing the final design and testing the application for stability and leaks.

[W17-W20] Writing the report
These weeks will be spent on writing and finalizing the report as well as preparing for the presentation and opposition.

C

Interview questions

C.1 Interview with SME at Jeppesen #1 questions

1. Vilka tjänster erbjuder ni till era kunder?
   (a) Hur mycket av produkten specialiseras för kunden?
   (b) vad säljer ni som tjänst och vad som komplett program?
2. Vad har Jeppesen för huvudkonkurrenter?
   (a) Ar dessa lika breda eller mer fokuserad på en sak?
3. Vilka tjänster har ni som vi skulle kunna conntecta till?
   (a) Väder
   (b) Förseningar
   (c) Gate information
4. Har ni några nya tjänster/produkter på ingeående?

C.2 Interview with SME at Jeppesen #2 questions

1. Vad driver piloter till att vara piloter?
   (a) Prestige
   (b) Frihet

   (c) Gillar ansvar/ha koll

2. Vad är det som är jobbigt med att vara pilot?
   (a) Vara reserv
   (b) Vidareutbildning/kontroller
   (c) Sova/vara på bortabas
   (d) Loggning av timmar
   (e) Oregelbudna arbetstimmar?
   (f) Behöver piloter vara insatta i många regelverk?
   (g) Fatigue
   (h) Säkerhet
3. Vilken sorts ubildning har piloter?
   (a) Skillnad mellan länder?
4. Hur är en vanlig pilot (stereotyper)
   (a) Tenkik insatta?
   (b) Evigt unga i sinnet?
   (c) Klädstil osv.
   (d) Köns-, etnicitet- och klassdistribution
5. Hur ser en vanlig dag ut för olika sorters piloter?
   (a) Senioritet och bra scheman
   (b) Hur får man sitt schema samt hur uppdateras det?
6. Haru nåt mer att tilläga som du tror är viktigt?

C.3 User interview questions

C.3.1 Administrativa angelägenheter

1. Hur ser ditt schema ut?
2. Hur långt interval är det över?
3. Hur får du det?
4. Vilket format är det i? (Är det digitalt?)

5. Hur mycket ändras schemat [under månaden]?
6. Hur säkert är det? [Vilken position/seniority har du?]
7. Hur långt fram “vet” man att schemat inte kommer ändras?

C.3.2 Verktyg

1. Importerar du ditt schema till någon kalender?
2. Vad använder du för vektyg?
3. Har du använt Rosterbuster?
4. Har du hört talas om det?
5. Vad är det för format?

C.3.3 Timezone

1. Vid flygschema fås alla tiderna i hemmatid?
2. Tänker man på vilken tidszon man befinner sig i just nu?
3. Är man “bättre” än gemene man på tidszoner?
4. Förbereda sig?
5. Förbereder du dig för de flighter du kommer köra?
6. Gör du något särskilt för att visualisera det för att få en känsla för de olika dagarna?
7. Finns det några ritualer?
8. Brukar du mentalt gå igenom den (flighten), säg på morgonen innan du går till jobbet?

Administrativa angelägenheter

1. Om uppdateringar behövs på schemat hur får du dem då?
2. Är det annorlunda om du är reserv / stand-by?
3. Hur får du gate info för dina flygningar?
4. (Är det på samma sätt som passagerarna eller finns det någon informationsportal?)
5. Är regulations något man tänker på i vardagen?
6. Behöver man kolla upp saker då och då?

7. Hur ofta behöver du uppdatera/repetera dina kunskaper om de regulations som påverkar dig?

C.3.4 Loggning

1. Loggar du flygdata?
2. Vilka sorters data? (flygtimmar, landningar, etc.)
3. Vad använder du för verktyg för att logga och vilket format sparar du all din logdata i?
4. (Finns det några reglering kring datan för att den skall kunna användas officielt?)
5. Har du vecko/månads rapporter eller liknande för din data?
6. Vad håller flygbolaget koll på åt dig?
7. Litar man på det?

C.3.5 Flygning - Teknisk inrikting

1. Använder ni iPads i jobbet (kanske i cockpit)?
2. Är de personliga eller gemensamma?
3. Loggar in på iCloud/AppleID för att personifiera upplevelsen?
4. Är de monterade i cockpit?
5. Hur tas detta emot av dig och kollegor? - Bra eller dåligt?
6. Hur ser användnigen ut privat?
7. Har du en smartphone?
8. Känner du att du“behärskar” den eller använder du den “bara”?
9. Kan man säga att detta en korrekt bild av piloters användningsmönster?
10. Änvänds de under flygnign?
11. Till vad? (Jobb eller underhållning).
12. Är de anpassade till detta på något sätt?
13. Kanske ljusstyrka eller dylikt?
14. Vad är det för stämming i cockpit? Är det så att man “bara umgås” (/gött snack) om inte något särskilt händer?
15. Vad gör man när man inte flyger? Eller när autopiloten kör?

16. Tex inflightsleep

Fatigue Risk Management

1. Har du hört talas om FRM?
2. Vad vet du om FRM?
3. Är det stor skillnad att landa på olika flygplatser? (Mer “utmattande”?)
4. Hur olika påfrästande är det att flyga under olika villkor?

CrewAlert

1. Har du hört talas om CrewAlert? (eller kanske använt det?)
2. Varför och till vad användes det?
3. Har du (eller de som använt det) några synpunkter?