****

**Team 52: Jaromir Latal & Lambros Zannettos**

**Client: The Mind Manifesto**

**(Imran Ahmad, Terry Hall)**

**Supervisor: Dr. Yun Fu**

****

**Table of Contents**

**1.** [**Table of Contents** 3](#_Toc448236586)

[**2. List of Figures** 4](#_Toc448236587)

[**3. Abstract** 5](#_Toc448236588)

[**4. Context** 5](#_Toc448236589)

[**4.1. Project Background** 5](#_Toc448236590)

[**4.2. Problem being solved** 5](#_Toc448236591)

[**5. Team Roles and Skills** 6](#_Toc448236592)

[**5.1. Jaromir Latal** 6](#_Toc448236593)

[**5.2. Lambros Zannettos** 6](#_Toc448236594)

[**6. Requirements** 7](#_Toc448236595)

[**6.1. Initial questions** 7](#_Toc448236596)

[**6.2. List of requirements** 7](#_Toc448236597)

[**6.3. Design** 8](#_Toc448236598)

[**6.4. Use Cases** 10](#_Toc448236599)

[**7. Work Distribution** 12](#_Toc448236600)

[**7.1. Jaromir Latal** 12](#_Toc448236601)

[**7.2. Lambros Zannettos** 12](#_Toc448236602)

[**7.3. Shared** 12](#_Toc448236603)

[**8. Technical Design** 13](#_Toc448236604)

[**8.1. Technology chosen** 13](#_Toc448236605)

[**8.2. Structure of the Application** 14](#_Toc448236606)

[**8.3. User Interface** 14](#_Toc448236607)

[**8.4. Design Patterns** 16](#_Toc448236608)

[**8.4.1. Singleton** 16](#_Toc448236609)

[**8.4.2. Template** 17](#_Toc448236610)

[**8.4.3. Command** 17](#_Toc448236611)

[**8.4.4. Observer** 17](#_Toc448236612)

[**8.5. External Software Libraries** 18](#_Toc448236613)

[**8.5.1. WebKit** 18](#_Toc448236614)

[WebKit is a layout engine for rendering web pages in web browsers. It is used in the web browser Safari, Chromium-based Google Chrome and Opera web browsers as well as implemented in almost all mobile web browsers. 18](#_Toc448236615)

[**8.5.2. SeekArc** 18](#_Toc448236616)

[**8.5.3. Google Analytics** 18](#_Toc448236617)

[**9. Usage** 19](#_Toc448236618)

[**9.1. Installation Requirements** 19](#_Toc448236619)

[**9.2. Building requirements** 19](#_Toc448236620)

[**9.3. Obtaining Google Analytics ID** 19](#_Toc448236621)

[**10. Testing** 20](#_Toc448236622)

[**10.1. Development Testing** 20](#_Toc448236623)

[**10.2. External Testing** 20](#_Toc448236624)

[**11. Evaluation and Conclusion** 21](#_Toc448236625)

[**11.1. Client Feedback** 21](#_Toc448236626)

[**11.1.1. Mid-development Feedback** 21](#_Toc448236627)

[**11.1.2. Pre-Final Feedback** 21](#_Toc448236628)

[**11.1.3. Final Feedback** 21](#_Toc448236629)

[**11.2. Project Evaluation** 21](#_Toc448236630)

[**12. Bibliography** 23](#_Toc448236631)

**2. List of Figures**

[**Figure 1:** UI sketches provided by client 8](file:///C:\Users\jermenkoo\Documents\GitHub\audava_meditation\Documents\team52_final_report.docx#_Toc448184590)

[**Figure 2:** UI sketch by our team 9](#_Toc448184591)

[**Figure 3:** Main differences between native and hybrid development (IBM, 2012) 13](#_Toc448184592)

[**Figure 4:** Splash screen during loading 14](file:///C:\Users\jermenkoo\Documents\GitHub\audava_meditation\Documents\team52_final_report.docx#_Toc448184594)

[**Figure 5:** Initial screen – media player 14](file:///C:\Users\jermenkoo\Documents\GitHub\audava_meditation\Documents\team52_final_report.docx#_Toc448184593)

[**Figure 6:** Application menu 15](file:///C:\Users\jermenkoo\Documents\GitHub\audava_meditation\Documents\team52_final_report.docx#_Toc448184595)

[**Figure 7:** Settings screen 15](file:///C:\Users\jermenkoo\Documents\GitHub\audava_meditation\Documents\team52_final_report.docx#_Toc448184596)

[**Figure 8:** Categories (All) screen 16](file:///C:\Users\jermenkoo\Documents\GitHub\audava_meditation\Documents\team52_final_report.docx#_Toc448184597)

[**Figure 9:** Categories (Relationships) screen 16](file:///C:\Users\jermenkoo\Documents\GitHub\audava_meditation\Documents\team52_final_report.docx#_Toc448184598)

# **3. Abstract**

The application we were asked to develop is meant to provide people escape from every day’s routine by providing them with multiple guided and unguided meditation courses (or “interventions”, as defined by client). The target user of this application is a busy person, who is likely to be overwhelmed and stressed; thus this application will allow him to slow down and relax. The application should be simple to use, including the UI and the user should be able to see how much time he has left and be able to jump to any place he wants. Another possible functionality is to provide user reminders / notifications that he has broken his meditation streak and should start over.

# **4. Context**

## **4.1. Project Background**

The Mind Manifesto (MM) is a set of toolboxes to help people improve their well-being and counter the negative effects of anxiety and stress in today’s hectic environment. The toolboxes are meditation tracks focused on certain areas of our lives, such as work, health or relationships – areas which have the greatest influence upon us.

Our clients, Imran Ahmad and Terry Hall, have asked us to create the mobile version of The Mind Manifesto, so that the interventions are available offline and on-the-go. This is meant to accompany the main version of The Mind Manifesto, which is currently available as a web application.

## **4.2. Problem being solved**

According to (Lu, 1999), “stress has become one of the most serious health issues of the twentieth century – a problem not just for individuals in terms of physical and mental disability, but for employers and governments who incur significant financial damage.” (Hassard, et al., 2014) reports that in year 2012 the costs calculated in the EU were €20 billion a year, a figure which has almost doubled since then; (Blanding, 2015) estimates the figure for US up to $190 billion dollars.

It has been thoroughly researched that meditation is effective at mitigating the effects of stress - (Black, Manocha, Sarris, & Stough, 2011) report that “meditation, …, is a safe and effective strategy for dealing with work stress and depressive feelings”; (Astin, 1997) concludes “techniques of mindfulness meditation, …, may represent a powerful cognitive behavioural coping strategy”.

This is in conjunction what our application is aimed at – to help people in high-stress work environments, such as doctors or lawyers, cope with stress by listening to the series of guided meditation tracks.

# **5. Team Roles and Skills**

## **5.1. Jaromir Latal**

* **Team Role:** Team Leader / Project Manager, Lead Interviewer for Requirements, Lead Tester
* **Key Skills:** I am proficient in Python, as it is the first programming language I have taught myself and have used it during multiple projects and programming competitions. I have also familiarity with C++ and Java due to taking International Baccalaureate Computer Science course.
* **Prior Project Experience:**
  + Machine Learning project in Python (scikit-learn) to predict stock prices
  + Patients Database in Java & MySQL for a child doctor
  + Educational Math Game written in Java using Swing GUI library
* **Communication & Teamwork:** Debating for 3 years have given me the opportunity to respect others’ opinions and approach them neutrally, finding the best possible solutions for multiple parties. Working as a Software Engineer in the largest agricultural software company in Slovakia provided me a plenty of opportunities to improve my teamwork and communication skills.

## **5.2. Lambros Zannettos**

* **Team Role:** Lead Software Engineer & UI Designer
* **Key Skills:** I am proficient in Android development/Java (Android Studio), as I have completed several online courses leading up to the App Project (at Coursera and Udemy). I have also become familiar with C, HTML and JavaScript as part of the course.
* **Prior Project Experience:**
  + Browser-based educational game to teach Python written in JavaScript and the EaselJS set of libraries.
  + Machine Learning project to predict susceptibility to UV radiation, using MS Azure ML Studio.
* **Communication & Teamwork:** Prior to this course, I was a full-time self-employed musician/composer for several years. The nature of my work was highly collaborative, as creative projects often are. This along with being part in many amateur and professional bands over the years, and my national service experience before that, has taught me to listen to others with as little personal bias as possible, and always strive to make the best decisions to benefit the project/end result.

# **6. Requirements**

After being introduced to the project & client by Dr. Yun Fu, we discussed with our team possible scopes of the application. For this we researched current applications on the market (for Android and iOS) in order to prepare ourselves better for the upcoming meeting with the client. Our client sent us specifically three to have a closer look at – *Calm, Headspace and Insight Timer*.

## **6.1. Initial questions**

During the first meeting with the client we asked following questions to know the size and the scope of the project, as well as any preferences client has for the platform / technologies and design.

* What is exactly the project about?
* What platform would you like to target? Are there any preferences?
* Are there any technologies / languages / frameworks we are expected / asked to use?
* Details about the application and its functionality:
  + Do you have any specific UI design in mind?
  + What features are required for the core functionality?
  + What are the additional features we might implement?
  + Should it be on-line / off-line application? (e.g. tracks stored remotely)
  + Should the application feature in-app purchases as well?
  + Should the user have some settings available?
  + How many songs do you plan the app to hold? (relevant for off-line version)

## **6.2. List of requirements**

Based on the client’s answers to the questions, we drafted a list of requirements in MoSCoW (Must-Should-Could-Would) style, splitting the requirements into 4 categories based on their importance for the success of the project. After meeting with HCI Teaching Assistant Aisling O’Kane we finalised it and have it approved by our client.

**Must have -** **What must be delivered, i.e. it is essential for this phase?**

* Capability to play mp3 files stored locally on the device, including common media functions like play/stop/pause.
  + Complete: Media player and all its functions work as intended.
* Ability to detect connected headphones as the content is to be listened to via headphones only.
  + Complete: Media player displays a message to the user that for best experience headphones should be plugged in.
* Minimal, simple and intuitive UI design.
  + Complete: The design was rated well by both client and users.

**Should have - What should be delivered as a high priority but not essential?**

* A reminder function by which the app reminds the user to take some time to meditate. This could be either at random times or at specific times each day.
* Skip forward/backward buttons that skip a set number of seconds.
  + Complete: The skip time in seconds can be set in settings.
* Remembering last place left off in a session (if it was closed before completion).
  + Incomplete: Removed after discussion with client – meditations are meant to be listened to from the beginning.
* A big countdown timer signifying the time left in a session.
  + Complete: The users is being shown the time left on the music player screen.

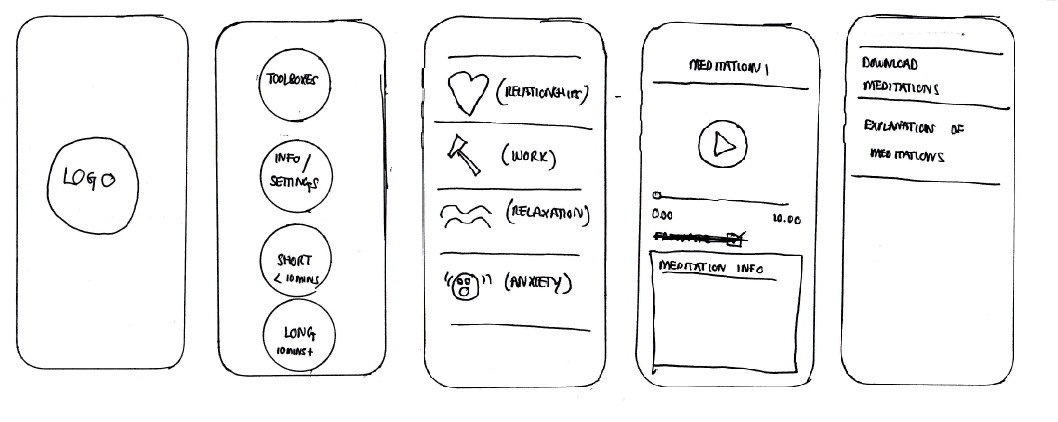
**Could have - What could be delivered if there was available time / budget / resource?**

* A store part of the app, where the users can buy and download new content.
  + Incomplete: Pushed back into next version after discussion with client.
* Translatable (this is easy to implement so could also belong in the Must list above).
  + Complete: Application is localised into user’s system language.

**Would have - What would be delivered if time / budget / resource was unlimited?**

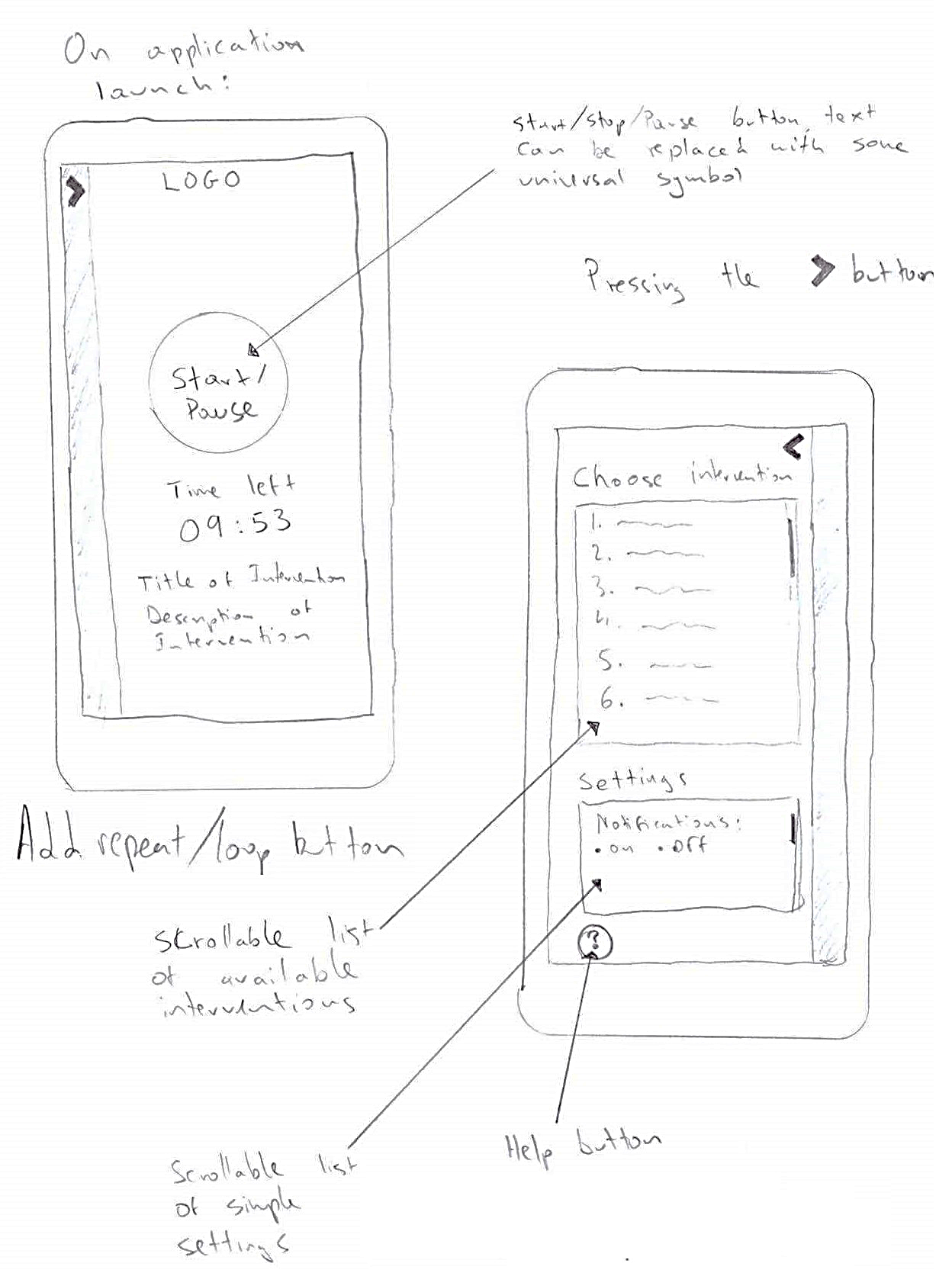
* Intelligent reminder that detects habits and breaks them.
  + Semi-complete: Not possible in the time-frame given, however logging for users’ action is present.
* Payment integration for new interventions in the application
  + Incomplete: Not enough time and issues present with testing in Google Play Store.

## **6.3. Design**

The client has asked us to deliver a minimal, simple and clean UI, so that the application will not be cluttered and every first-time user will immediately know where to click. We created a functional UI prototype independently and compared it to the client’s sketches, settling on the final design.

**Figure 1:** UI sketches provided by client

(from left: splash screen, main menu, toolboxes, media player, settings)



**Figure 2:** UI sketch by our team

Both UI mock-ups are quite similar **–** the menu is hidden from the user not to obstruct his view and design of media player shares the idea of a big button along with time remaining being shown in the session. We agreed on adding settings screen as a separate option in the menu rather than having shown all the settings there. Based on this we settled on the final design, which is shown later in this report (Section 8.3), already implemented.

## **6.4. Use Cases**

We carried out use case analysis in our project in order to better understand the system we will be building. It also allowed us to further look at our requirements and whether they can be refined better. We did so by illustrating interaction between the system (application) and its only actor – user.

|  |  |
| --- | --- |
| Use Case (UC1) – Application is opened for the first time | |
| Primary Actor | User |
| Precondition | The application has not been opened yet. |
| Postcondition | The main menu appears and the tutorial screen will not appear anymore. |
| Main paths | 1. User opens the application. 2. Splash screen appears for a few seconds. 3. Brief tutorial appears describing the application to the user and how to navigate in the application. |
| Alternative paths | None |

|  |  |
| --- | --- |
| Use Case (UC2) – Application is opened | |
| Primary Actor | User |
| Precondition | The application has not been running. |
| Postcondition | The main menu appears. |
| Main paths | 1. User opens the application. 2. Splash screen appears for a few seconds. 3. The main menu appears. |
| Alternative paths | None |

|  |  |
| --- | --- |
| Use Case (UC3) – A new intervention is chosen | |
| Primary Actor | User |
| Precondition | The application is running. |
| Postcondition | The chosen intervention starts playing. |
| Main paths | 1. User selects the intervention’s category. 2. User selects a specific intervention. 3. The menu is hidden and the player screen appears. 4. The chosen intervention starts playing. |
| Alternative path #1 | 4. a) The intervention track cannot be loaded.  4. b) User is notified and returned back into the selection screen. |

|  |  |
| --- | --- |
| Use Case (UC4) – Intervention is paused or resumed | |
| Primary Actor | User |
| Precondition | An intervention has been loaded successfully and is playing. |
| Postcondition | An intervention is either paused or resumed. |
| Main paths | 1. User selects the pause button in the player screen. 2. The intervention stops playing. |
| Alternative paths | 2. a) The intervention has been paused previously.  2. b) The intervention is resumed. |

|  |  |
| --- | --- |
| Use Case (UC5) – Skipping to a certain part of intervention by scrubbing | |
| Primary Actor | User. |
| Precondition | The media player screen is active and an intervention is loaded. |
| Postcondition | The time position of the track is changed. |
| Main paths | 1. User scrubs on the position slider using a finger. 2. Once the finger is released, the track jumps to the position. 3. Playback continues from the new position |
| Alternative paths | 3. a) The intervention was paused before scrubbing. The playback does not continue until resumed. |

# **7. Work Distribution**

## **7.1. Jaromir Latal**

* Add error handing
  + Complete: Error handling has been added to prevent user from seeing errors / notifications; all errors are being logged.
* Add usage logging
  + Complete: Google Analytics for Android has been used to monitor users’ usage of the application.
* Fix bugs and issues found during the development
  + Complete: All reported issues on GitHub have been closed after being successfully solved.
* Refactor code
  + Complete: During the development we strictly adhered to OOP principles and made use of multiple design patterns.
* Test the application
  + Complete: The application has been subjected to multiple types of testing, including the issues and bugs found during development.

## **7.2. Lambros Zannettos**

* Develop music player functionality
  + Complete: Implemented media player from scratch with cueing by desired time in seconds.
* Intelligent intervention filtering based on the category
  + Complete: After a category is chosen user sees only interventions belonging to that certain category.
* Getting metadata from interventions
  + Complete: Information about intervention are being read from an ID3 tag in the media file instead of being somewhere hardcoded.
* Changeable settings
  + Complete: User’s settings are saved persistently on the phone.

## **7.3. Shared**

* Produce bi-weekly reports
  + Complete: Bi-weekly reports have been filled in and submitted on time.
* Create an initial UI mockup
  + Complete: UI mockup has been completed based on the individual mockups created by each of us.

# **8. Technical Design**

## **8.1. Technology chosen**

After having agreed with the client that we want to target **Android** platform due to number of devices on the market, we had to decide which technology to use. We considered the following two possibilities: **Android Studio** (native Android SDK – Software Development Kit) and **3rd party SDKs** (such as **Cordova** or **Xamarin**).

**Android Studio** is the IDE which provides access to Android SDK - a modified version of Java including Android features and Google APIs - used to develop native Android applications.

**Xamarin** is a cross-platform mobile development tool used to develop applications both for Android and iOS (in C#). Xamarin apps consist of two layers:

1. **iOS / Android** – containing the UI and application layers (written in C#)
2. **Core Library** – containing shared code such as business logic

**Cordova** is a mobile cross-platform application development framework for **Android**, **iOS**, **Windows** **Phone** and more. Development is done in HTML5, CSS3 and Javascript instead of platform specific APIs (such as the Android SDK), however it allows using native device APIs providing access to device’s camera, file system and more.



**Figure 3:** Main differences between native and hybrid development (IBM, 2012)

We decided to utilise the native **Android SDK**. The reasons were the following:

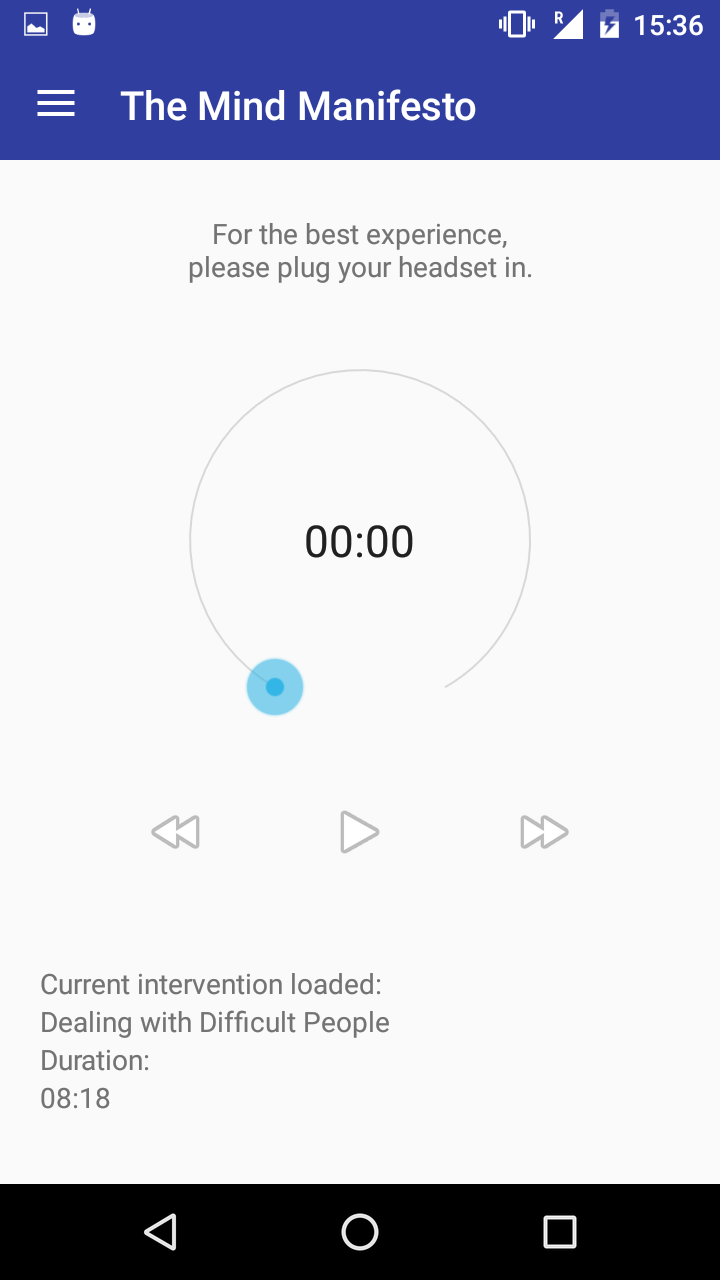
1. Using native APIs guarantees the look-and-feel will be the same on every device for the given platform.
2. Since the application is meant to be used for a longer amount of time (for meditation), the web application could run out of the memory or be less responsive (O'Dell, 2013), possibly resulting in bad user feedback.
3. Going native guarantees future compatibility, as Android Studio always adapts to the newest **Android SDK**, however when a new version of Android is released, cross-platform development tools do not support it until an update is released.

## **8.2. Structure of the Application**

Flowchart

Code hierarchy

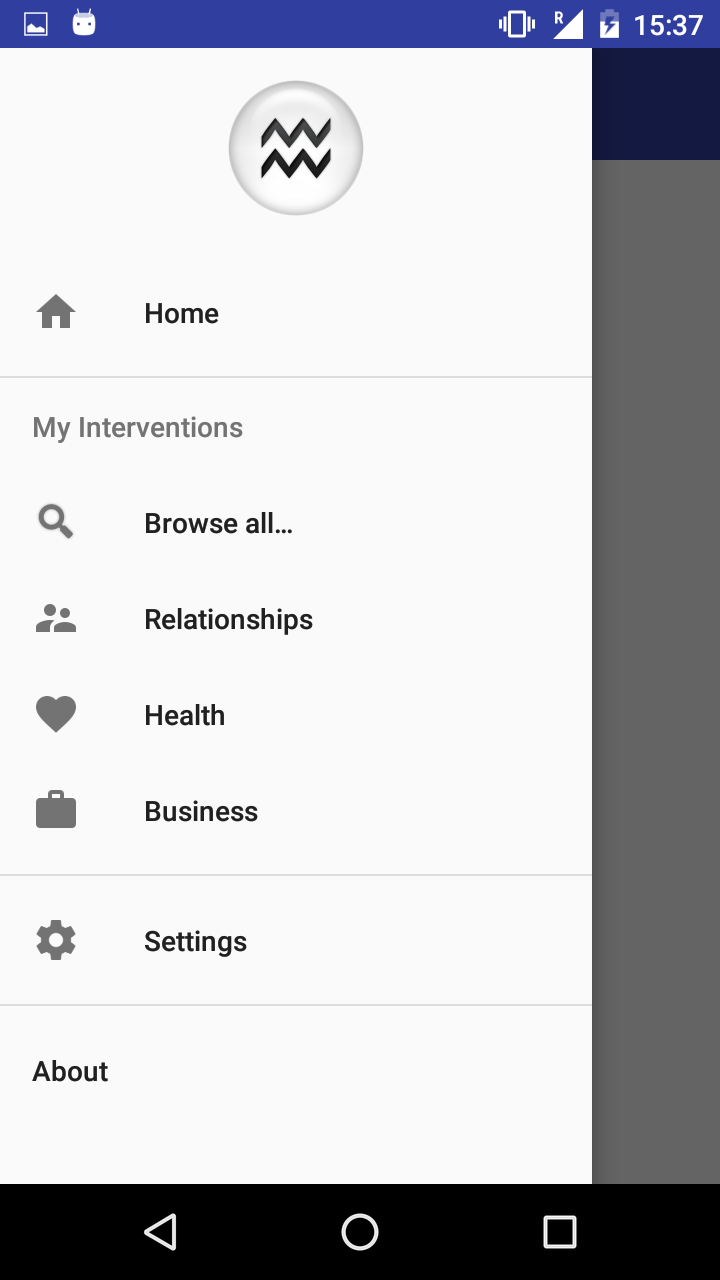
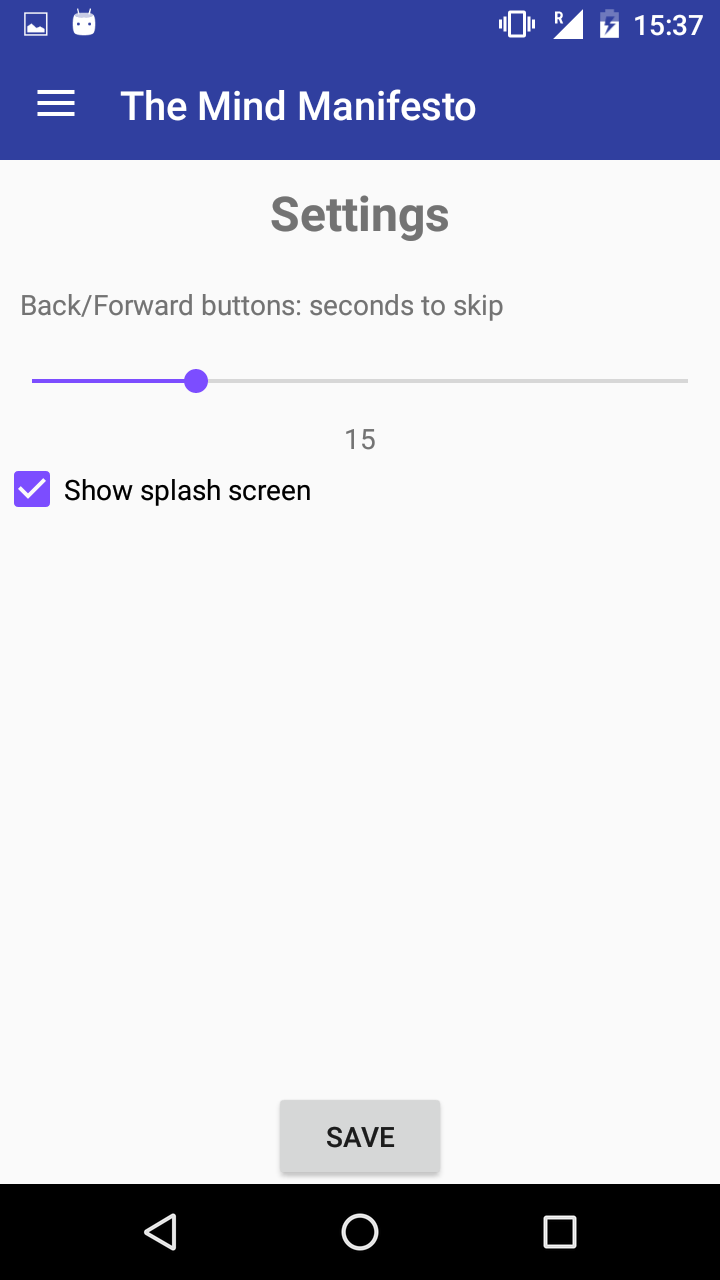
## **8.3. User Interface**

We designed our user interface so that it is intuitive to use – it consists out of simple design blocks (buttons, sliders, textviews) composed together for the best user experience. We aimed the app to be easy to use for everybody without having to read a manual or spend a long time in the application, discovering all its features.

**Figure 5:** Initial screen – media player

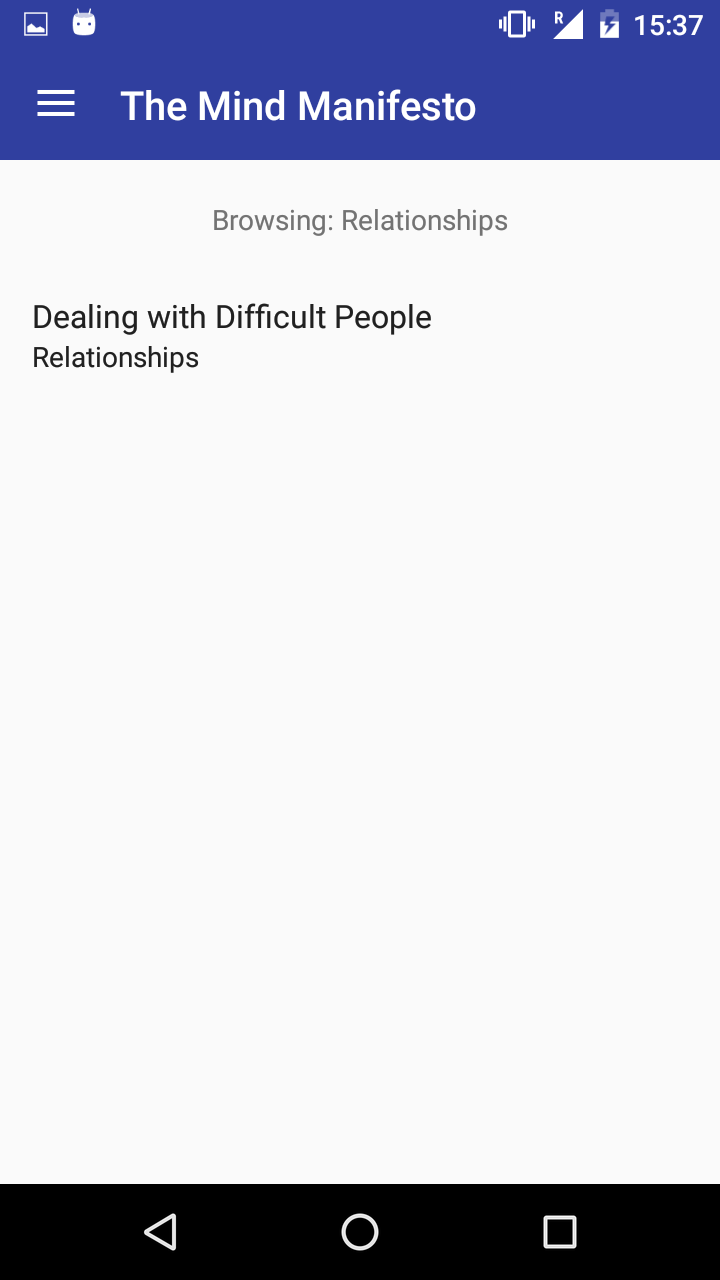
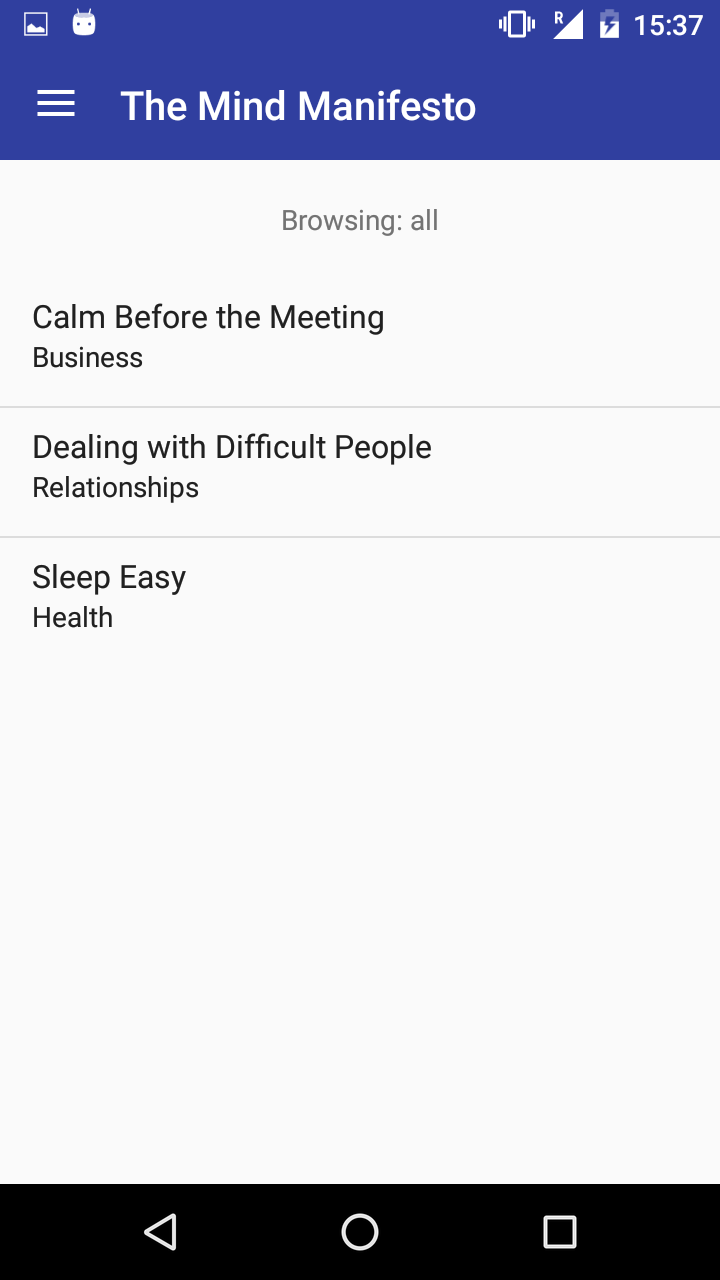
When the application is launched, the splash screen is shown (Figure 2) during which the files needed for application launch are being loaded. Afterwards the user is shown directly the media player (Figure 3) with one of the meditation tracks loaded, so that he can try out the application directly.

**Figure 4:** Splash screen during loading

The menu can be accessed by swiping from the right edge of the screen. It contains basic options, namely main screen (where the media player is located), interventions available in the current version (split into their own categories), settings screen in which the user can alter the amount of time he wants to cue forward or backward and whether the splash screen should be shown.

**Figure 6:** Application menu

**Figure 7:** Settings screen

In order to play an intervention, user has to choose the category he wants to listen to and then choose a specific intervention out of the list. There are two types of categories: all, where all the interventions in the application are available or specific, such as Relationships, Health or Business, where only interventions belonging to this specific category are shown.

**Figure 8:** Categories (All) screen

**Figure 9:** Categories (Relationships) screen

## **8.4. Design Patterns**

### **8.4.1. Singleton**

The Singleton pattern restricts the instantiation of the class to only one object, meaning only one instance of a class can be created, often being accessed globally.

Singleton has been implemented in our application in the MediaPlayerSingleton class. We decided to use so-called eager initialisation of Singleton, because our application will always need an instance (so that the MediaPlayer can play interventions in the applications). Moreover, the cost of creating one instance is not too expensive, which further affirmed our decision.

Using this design pattern ensured that the MediaPlayer instance always exists and that after changing screens the reference to the instance still exists, which is better rather than creating a completely new instance of MediaPlayer when a different meditation track is chosen.

### **8.4.2. Template**

Template pattern is a behavioural pattern design, which defines the skeleton of the algorithm and defers some steps to the subclasses – they can re-define certain steps as needed without changing the algorithm’s structure (which is common for all of them).

We made use of this pattern when creating items belonging to each category. While each category should have different items (some of them can be the same, e.g. a meditation track will be both in its own category and in all categories), the behaviour should be the same – when clicked the music player screen should be shown and the song played.

Usage of this design pattern allowed us to remove duplicate code which would have been otherwise needed for every item in the menu.

### **8.4.3. Observer**

The Observer pattern is a behavioural pattern - when an object is modified, all other object depending on it (called observers) are notified automatically by calling one of their methods.

Observer pattern in our application is implemented in the HeadphoneStateReceiver class, which extends the built-in BroadcastReceiver. Once the headphones are plugged in, the HeadphoneStateReceiver receives notification from ACTION\_HEADSET\_PLUG listener and shows a notification to the user – to notify him that the headphones have been plugged.

## **8.5. External Software Libraries**

During the development we used multiple 3rd party libraries as they provided functionality which could not obtained otherwise. Its usage allowed us to refine performance of the application as well as improve its design and usability.

### **8.5.1. WebKit**

### WebKit is a layout engine for rendering web pages in web browsers. It is used in the web browser Safari, Chromium-based Google Chrome and Opera web browsers as well as implemented in almost all mobile web browsers.

We used WebKit in order to be able to serve static *.html* files from our application - About page is stored as a persistent *.html* file which is shown using Android’s WebView, which is View responsible for displaying webpages.

### **8.5.2. SeekArc**

SeekBar is an extension of ProgressBar that adds a dragable element, which can be used to adjust the current progress level. SeekArc is 3rd party library which extends the SeekBar – it wraps the SeekBar around in a circle.

We originally used SeekBar to allow users to change the current position in the song by scrubbing, however we decided to rather use the 3rd party SeekArc library. We had multiple reasons:

* SeekArc looks aesthetically much more pleasing than SeekBar and fills space better than SeekBar.
* SeekArc has resulted in better performance than SeekBar – after benchmarking we discovered updating song’s position is smoother and without any stuttering.

Since the library used already deprecated API, we modified the library to adhere to the newest API available, making use of Android Support Library to support older devices.

### **8.5.3. Google Analytics**

Google Analytics is an analytics service offered by Google that tracks and reports website traffic. It is provided in two versions – one of them Google Analytics for Mobile Apps, an SDK allowing gathering usage data from Android and iOS.

Google Analytics has been used to track and collect data about users’ behaviour in the application, providing valuable information about how they use the application – based on the data it can be decided what categories are liked the most and the least, what meditation tracks are most popular – which can be then used to create content for future versions and provide tailored user experience.

We decided to choose Google Analytics instead of 3rd party tracking providers because Google Analytics is free up to 10 million requests / month and its integration into the application is much easier than using 3rd party providers, as Google provides their own library with all the documentation.

# **9. Usage**

## **9.1. Installation Requirements**

The application can be installed on any Android device whose Android version is at least 4.0.3 (Ice Cream Sandwich) or higher. We decided to support all devices from this version onwards to have the highest market share possible (4.0.3 and newer devices account for 97.3% all Android devices (Google, 2016)), while making use of the newest features and libraries in the Android SDK.

Since the application uses features available only for Android 5.0 and newer, we make use of official Android Support Library in order to maintain compatibility with devices with Android version 4.0.3-4.4. (Google, 2016)

## **9.2. Building requirements**

In order to build the application, simply open either *audava\_meditation.iml* or *TheMindset.iml* using Android Studio. Once the project is loaded, the application should be built automatically using Gradle build system, which will download and compile all the dependencies of the application and run tests. Provided the testing has finished without any error, an *.apk* (Android package) file is built, which can be then installed and used on any supported Android device (outlined above).

The Gradle build script generates two *.apk* files in different configurations - Debug and Release. Debug build is a larger binary containing debugging symbols to make finding bugs and errors, as it allows getting log output from the application / device (Barth, 2015), while Release build disables getting the logs in order to harden the application from Reverse Engineering. (analysing the compiled binary to obtain its source code)

Release build differs mainly in two things: it is also smaller, as unnecessary code is stripped off and the application can be signed with own certificate to ensure that the authorship can be verified and that any subsequent upgrades will be still installed as one package. (Google, 2016)

## **9.3. Obtaining Google Analytics ID**

This application uses Google Analytics for Mobile Apps in order to track and gather data about users’ behaviour. To enable tracking within the application, a configuration file needs to be generated at [Google Developers - Enable Google services for your app](https://developers.google.com/mobile/add?platform=android).

After the application name and application’s package ID have been entered, user is prompted to choose services he would like to add to his app. Google Analytics should be ticked and user should continue to *Generate configuration files*, where configuration file *google-services.json* can be downloaded.

The last step is to copy the *google-services.json* file to *app/* directory – afterwards the tracking will be functional after the application has been rebuilt.

# **10. Testing**

## **10.1. Development Testing**

During development we tested our application on multiple devices to ensure that it works as needed. We used following devices during testing (sorted by their Android version):

* Huawei Ascend, model number: Y330, OS version: Android 4.2
* Sony Xperia Z3 Compact, model number D5803, OS version: Android 5.1.1
* Motorola Moto G2, model number: XT 1068, OS version: Android 6.0

We chose variety of devices with different OS versions and screen resolutions to test the application on the widest range of devices as possible – Huawei Ascend has Android 4.2, which meant we could test whether the compatibility provided by Android Support Library is maintained.

Different screen resolutions were important for checking whether the application and all the graphic assets and UI elements (such as TextViews, SeekBars, Buttons) scale on different screen sizes to provide the best User Experience possible.

## **10.2. External Testing**

During and after development we conducted tests throughout our development, asking third party people with no previous knowledge about the application to try it out and provide their opinion – score out of ten in following categories: Overall, User Interface, User Experience, Ease of Use and Problem Encountered.

*Few words about rating*

# **11. Evaluation and Conclusion**

## **11.1. Client Feedback**

During the course of the development we stayed in touch with our client in order to update him on any progress as well as ask for clarifications on questions we had. We met with him few times to demonstrate him the current prototype version of the application and ask him for his feedback.

### **11.1.1. Mid-development Feedback**

At the beginning of the March we met with the client to show him the latest version of the application – it lacked few features, such as being able to choose different interventions (we had only one from the client at that time), otherwise it was fully functional.

The client feedback was positive, they were satisfied with both the design and the functionality of the application. We were notified that the name of the application has changed from **The Mind Set** to **The Mind Manifesto** and that we will get the new graphic sprites as soon as possible.

### **11.1.2. Pre-Final Feedback**

At the beginning of April we sent the client the pre-final version of the application so that they are able to test it in detail if it conforms to all their requirements.

The client feedback was mostly positive – we were said the application is looking good and that they are looking forward to the final result, however there were handful of changes to be made – we received the logo to be used for the splash screen, along with the colour scheme the application should have, the names of the categories should be changed to Business, Health and Relationships and we got few example interventions to be implemented in the application.

### **11.1.3. Final Feedback**

In the middle of April we met with the client to demonstrate him the final version of the application, implementing all the important features. We planned the final meeting to be approximately two weeks before the deadline so that we implement the changes will have received from the client as well as fix any bugs or issues noticed.

## **11.2. Project Evaluation**

The aim of our project was to develop a guided meditation application for Android from scratch. Judging the success of the project as a whole, the project can be declared as successful and the product built as fully functional and the application released as version 1.0.

Nonetheless there is a number of future improvements, such as:

* **In-App purchases through Google Play Store** – adding in-app purchases would be beneficial, as it would be the main revenue generating element for the application (provided it will remain free and without any ads). Moreover, In-App purchase. Moreover, In-App purchases would allow the application size to shrink when installed as interventions could be downloaded remotely.
* **Remote storage of interventions** – interventions present in the application could be stored remotely on a cloud storage (such as Amazon EC2 or Google Cloud), therefore the application size would be small when installed for the first time. However, in that case internet connection would be needed when launching the application for the first time in order to download the interventions into the device.

# **12. Bibliography**

Astin, J. (1997). Stress Reduction through Mindfulness Meditation. *Psychotherapy and Psychosomatics, 66*(2), 97-106. doi:10.1159/000289116

Barth, S. (2015, December 6). *What is the difference between a debugged version android apk and a release non-debugged version android apk?* Retrieved April 11, 2016, from StackOverflow: https://stackoverflow.com/revisions/34116685/2

Black, D., Manocha, R., Sarris, J., & Stough, C. (2011). A Randomized, Controlled Trial of Meditation for Work Stress, Anxiety and Depressed Mood in Full-Time Workers. *Evidence-Based Complementary and Alternative Medicine*, 1-9. doi:10.1155/2011/960583

Blanding, M. (2015, January 26). *Workplace Stress Responsible For Up To $190B In Annual U.S. Healthcare Costs*. Retrieved April 8, 2016, from Forbes: http://www.forbes.com/sites/hbsworkingknowledge/2015/01/26/workplace-stress-responsible-for-up-to-190-billion-in-annual-u-s-heathcare-costs/#730ba2444333

Google. (2016, March 29). *Dashboards*. Retrieved from Android Developers: https://developer.android.com/about/dashboards/index.html

Google. (2016, March 29). *Maintaining Compatibility*. Retrieved from Android Developers: https://developer.android.com/training/material/compatibility.html

Google. (2016, April 11). *Signing Your Applications*. Retrieved from Android Developers: https://developer.android.com/tools/publishing/app-signing.html

Hassard, J., Teoh, K., Cox, T., Dewe, P., Cosmar, M., Grundler, R., . . . Van den Broek, K. (2014). *Calculating the Cost of Work-related Stress and Psychosocial Risks.* Luxembourg: Publications Office of the European Union. doi:10.2802/20493

IBM. (2012, April). *Native, web or hybrid mobile-app development*. Retrieved March 30, 2016, from IBM: ftp://public.dhe.ibm.com/software/pdf/mobile-enterprise/WSW14182USEN.pdf

Lu, L. (1999, June). Work Motivation, Job Stress and Employees' Well-Being. *Journal of Applied Management Studies, 8*(1), 61-72.

# **Appendices**

## **Appendix A – Minutes from Meetings**

## **A1 Bi-Weekly Report #1**

**Overview:**

During the first two weeks we established roles in the team, had an initial meeting with client and conducted market research on already existing solutions. Based on the latter two we drafted and finalised MoSCoW requirements. We went through initial UI design and discussed it with the client.

**Meetings conducted:**

**22nd January - Meeting with team Topic: Market Research**

We met briefly with my team partner and we discussed the possible scopes of the application after being introduced to the project by Yun Fu. We are supposed to create a guided meditation application – while I do not have any experience, my partner already has and showed me some of them on his phone.

Furthermore we decided to research the current apps available on the market (web, Android and iOS) in order to prepare ourselves better for the upcoming meeting with client – Jaromir looked at iOS and web applications, while Lambros at Android ones.

**25th January - Meeting with team Topic: Market Research**

For preparation we went over our research on the already existing solutions plus we took a detailed look at those our client sent us – *Calm, Headspace and Insight Timer*.

**27th January - Meeting with client Topic: Initial Client Meeting (Application)**

We had our initial meeting with our client in which we were introduced to the project in more detailed manner, including scope, users, requirements and its goal. We have been requested to build a guided meditation application for smartphones. We discussed client’s ideas in conjunction with our backgrounds in order to be able to leverage our skillset most efficiently and decided on Android as it has more users than iOS plus development for Android is available without having access to a device with Mac OS X.

We discussed possible features of the application and its design and agreed on exchanging possible UI mock-ups next week (1st – 7th February). As for future communication & contact, we agreed on using e-mail to contact each other and meet whenever possible and required.

**2nd February - Meeting with team Topic: UI mockups**

As suggested by HCI Teaching Assistant Aisling O’Kane, we prepared some UI mockups and compared them. These and namely the ones we will received from client will be used to design the UI of the application.

**5th February – Meeting with team Topic: Discussion of MoSCoW requirements**

This meeting was primarily concerned with the discussion of the MoSCoW requirements and preparing their final version for the client. This was later sent to him in order to edit in any changes discussed and agreed on and to be able to finalise the report.

**10th February – Meeting with Aisling O’Kane Topic: Finalise the MoSCoW and UI**

At this meeting we discussed our progress so far, namely regarding the MoSCoW requirements report and UI mock-ups (both from client and ours). The Teaching Assistant confirmed that we are on the right track and that from now on we should identify the use cases in our project (based on the UI mock-ups) and common variables and data dictionaries.

**Tasks completed**

|  |  |  |
| --- | --- | --- |
| **Number** | **Task** | **Status** |
| 1. | Research on already existing applications | Done |
| 2. | Initial meeting with the client | Done |
| 3. | Drafted & Finalised MoSCoW report | Done |
| 4. | Created UI mock-ups and compared them to client’s ones | Done |
| 5. | Discussed our progress with HCI Teaching Assistant Aisling O’Kane | Done |

**Jaromir Latal**

During the initial period I researched iOS and web competitors present on the market. I created the draft of MoSCoW report and finalised it using client’s feedback. I was responsible for taking minutes this week.

**Lambros Zannettos**

During the first two weeks I had a look at Android competitors present on the market, creating a report presented to the client about the prices and the popularity of the applications. Furthermore I started to look into developing applications Android in order to prepare for the upcoming development.

**Plan for the next 2 weeks**

|  |  |  |
| --- | --- | --- |
| **Number** | **Task** | **Owner** |
| 1. | Research into technologies available | Jaromir |
| 2. | Create the final mock-up of app’s UI | Lambros |
| 3. | Finalise the project requirements | Both |
| 4. | Identify classes and use-cases needed | Both |
| 5. | Request sample recordings from the client | Jaromir |

## **A2 Bi-Weekly Report #2**

**Overview:**

These two weeks have been very productive. We researched all the possible development options and settled on using Android Studio and Java. We started creating the UI for the app and the basic functionality like menus. We also researched how some of the other Must-Have features can be implemented, like Media Player functionality, timers and saving settings.

**Meetings conducted:**

**15th February – Skype meeting with team Topic: Planning**

After having finalized the MoSCoW requirements at the end of last week, this meeting was used to discuss our plan for how to proceed with the project. We recapped what we need to achieve, and the methods with which we decided to approach this, and organised the resources (online lessons/guides/IDEs) required to do so. We decided to individually start experimenting with Android Studio and to keep each other up to date with our progress.

**17th February – Skype meeting with team Topic: Progress**

Last meeting was all about planning the next steps. This brief meeting was a catch-up, and we used the time to show each other what we have learned and any new thoughts we had about the process of development.

**19th February – Skype meeting with team Topic: Progress II**

Another progress report meeting. We both had enough time to explore Android Studio and start experimenting with different features and methods. This means that we have each identified potential difficulties and so much of this meeting was about preparing to tackle those issues.

**22nd February - Meeting with team Topic: Final UI discussion**

Based on the UI drafts that the client sent, and ideas discussed in the original client meeting, in this meeting we discussed options for the UI. The main issue discussed was the placement of the option menu. We have settled on a sliding menu that slides in form the left, over the main screen. This is not necessarily the final menu design, but implementing it then makes it easy to change it into another format later depending on client feedback.

**25th February - Meeting with team Topic: Use-Cases discussion**

We discussed and created several use cases to guide us in the creation of the first prototype. These describe basic functionality and navigation, and are based on our discussions with the client and our own research into other similar apps currently on the market.

**26th February – Meeting with team Topic: Classes**

In this meeting, we created an outline of the functionality of the app using class diagrams/UML. This was really useful, as looking at the app broken down into smaller constituents gave us an even clearer view of what needs to be done and how to efficiently divide the workload between us.

**Tasks completed**

|  |  |  |
| --- | --- | --- |
| **Number** | **Task** | **Owner** |
| 1. | Research into technologies available | Jaromir |
| 2. | Create the final mock-up of app’s UI | Lambros |
| 3. | Finalise the project requirements | Both |
| 4. | Identify classes and use-cases needed | Both |
| 5. | Request sample recordings from the client | Jaromir |

**Jaromir Latal**

I researched the technologies available for the development and researched sample recording from the client which will be used for the prototype demonstration. In addition I worked on identifying classes and use-cases needed.

**Lambros Zannettos**

I have spent the better part of these two weeks learning about Android development using Android Studio, via Lynda and Udemy courses, and also YouTube lessons for more specific features. I have put together a final mock-up of the app’s UI which includes a sliding menu and a so-called “hamburger” button on the top left, as an extra way for the user to reach the menu. I researched libraries required to add media player functionality and started learning about timers. I was responsible for taking minutes this week.

**Plan for the next 2 weeks**

|  |  |  |
| --- | --- | --- |
| **Number** | **Task** | **Owner** |
| 1. | Add media player functionality | Both |
| 2. | Add more menu options | Both |
| 3. | Finish the prototype | Both |
| 4. | Present the prototype to UCL advances | Both |
| 5. | Request the sprites from the client | Jaro |

## **A3 Bi-Weekly Report #3**

**Overview:**

During this week we made a significant progress on the development – we finished the media player functionality, added more menu options and most importantly – received feedback both from client and supervisor what to improve on. During next week we plan to tackle major usability functionalities, such as auto-scaling on all displays and robust error handling.

**Meetings conducted:**

**29th February – Meeting with team Topic: Development planning**

At this meeting we assigned roles during the development – Lambros will work on the media player functionality while I will work on translating the application into different languages. We decided on employing pair programming during allocated lab sessions so that we can solve problems easier.

**2nd March – Skype meeting with team Topic: Progress**

During this meeting we discussed the progress so far and set the goals we want to finish within this week. We planned what to tackle during next week – prepare for the demo of the application to our client (schedule a meeting) and our supervisor.

**3rd March – Meeting with team Topic: Development**

During this meeting we joined the allocated lab session in which we continued the development of the product – at the moment we are implementing the media player functionality into the application to enable implementation of the interventions. We were supposed to present the prototype but it has been postponed for next week.

**9th March - Meeting with client Topic: Progress discussion, demo**

During this meeting we demoed the application to our client – they were extremely satisfied, quoting: “this is exactly we wanted” and “nice, simple and clean UI.” We agreed on sending them the .apk of the current version to provide us more feedback after using it and discussed the size of the media files - agreed on compressing them as much as possible. Lastly we have been informed that the name has changed to “The Mind Manifesto” and they will send us the graphic resources (sprites) as soon as possible.

**10th March - Meeting with team Topic: Development & Feedback**

During this meeting we showed the first prototype of the application and received feedback what to improve. We were advised by our supervisor to remind the client to send us the graphic resources and then to make a Gantt chart to be able to track our progress with development over time.

**11th March – Meeting with team Topic: Further Improvements**

Today we discussed the current state of the application and worked on making it more modular – so that it is expandable in the future. We will take a look at the size of media files as well to find a suitable alternative towards the large size of the .apk file.

**Tasks completed**

|  |  |  |
| --- | --- | --- |
| **Number** | **Task** | **Owner** |
| 1. | Add media player functionality | Both |
| 2. | Add more menu options | Both |
| 3. | Finish the prototype | Both |
| 4. | Present the prototype to UCL advances | Both |
| 5. | Request the sprites from the client | Jaro |

**Jaromir Latal**

During this week I worked on polishing codebase and adding more menu options. I also researched error handling for our application. In order to manage our progress better, I started making a Gantt chart describing the current development goals and their status. I was also responsible for taking minutes.

**Lambros Zannettos**

I worked on making media player functional within the app – created the barebones version, added automatic timestamp update and rewind/forward by 15 seconds. I created basic graphic sprites for the application for the demo presentation.

**Plan for the next 2 weeks**

|  |  |  |
| --- | --- | --- |
| **Number** | **Task** | **Owner** |
| 1. | Finish functionality of the menu | Lambros |
| 2. | Ability to list all existing interventions in the application | Lambros |
| 3. | Make the application scalable on all resolutions | Jaro |
| 4. | Add error handling | Jaro |
| 5. | Remove duplicate code | Both |

## **A4 Bi-Weekly Report #4**

**Overview:**

This week saw a significant improvement in how the app works internally. We implemented another design pattern, the Singleton, which allowed us to ensure that only one instance of the *MediaPlayer* object exists at any given time; this makes sure the state is saved and that no unnecessary resources are held up. Use logging, error handling and multiple media file functionality are also things we worked on this week.

**Meetings conducted:**

**14th March – Meeting with team Topic: Progress**

Continued to work on making different parts of the code more independent from each other. Extracted some functions and removed constant from code and into a constants class to better keep track of them.

**17th March – Meeting with team Topic: Progress/Bug fixes**

Worked on removing some bugs that had come up by testing on physical devices. These included things like the media player becoming disconnected form the media file it was handling.

**18th March – Meeting with team Topic: Progress with menu**

We managed to make the menu a little bit smarter, by making it check whether the menu option selected was actually the activity currently on display, and if it is, to just close the menu rather than reloading the activity needlessly. This immediately made the app run smoother.

**21st March – Meeting with team Topic: Logging/Persistent data**

Today we looked into different options for saving persistent data on Android. This will be relevant for us, not only for keeping the logs of the usage of the app, but also for saving settings, the state of the media player etc.

**22th March - Meeting with team Topic: Singleton Pattern/Logging**

Today we finalised the way the media player runs in the app. By using the singleton pattern the media player object is now instantiated only once and used across the lifespan of the app. This way it never loses its state. We also started working on logging functionality to provide stats for the client.

**24th March – Meeting with team Topic: Logging/Settings**

In today’s meeting, we continued work on action logging and started working on a mechanism to handle multiple media files. As media files will potentially be streamed from a server in the future, we need to make this feature easy to plug in to for future developers.

**Tasks completed**

|  |  |  |
| --- | --- | --- |
| **Number** | **Task** | **Owner** |
| 1. | Implemented the Singleton pattern for the media player | Lambros |
| 2. | Started working on multiple intervention functionality | Lambros |
| 3. | Further improved robustness of error handling | Jaro |
| 4. | Started working on usage logging for client to analyse | Jaro |
| 5. | Tested – removed bugs – tested – removed bugs… | Both |

**Jaromir Latal**

During this week I worked on improving the error handling within the application, making sure all exceptions are processed gracefully and do not interfere with the functionality of the application. I researched on methods how to monitor usage of the application – which parts of the application are most popular, are used least often – characteristics describing how users interact with the application.

**Lambros Zannettos**

After reading up on the Singleton design pattern, I managed to implement it for the media player functionality of the app. I also started experimenting with different ways of handling multiple files.

**Plan for the next 2 weeks**

|  |  |  |
| --- | --- | --- |
| **Number** | **Task** | **Owner** |
| 1. | Continue working on logging | Jaro |
| 2. | Make preferences persistent/add settings screen | Lambros |
| 3. | Work on final report | Jaro |
| 4. | Implement ID3 reading capabilities (optional) | Lambros |
| 5. | Further encapsulate code | Both |

## **Appendix B – Client Communication**