Diary Application
Microsoft Band Monitor

Computer Science
School of Mathematical and Computer Sciences

Dissertation submitted as part of the requirements for the award of the degree of MSc in Information Technology (Software Systems)

Name: Luis Alberto Gutierrez Iglesias
Person ID: H00237078

Supervisor: Ruth Aylett
27 August 2016
Declaration

I, Luis Alberto Gutiérrez Iglesias, confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed:  

Luis Alberto Gutierrez Iglesias

Date:  

Friday, 27th of August, 2016
Abstract

A diary is commonly defined as a book where people write personal experiences and events sorted by date. Frequently used as a tool for self-awareness, it also allows users to remember past events, possibly associated with people or other relevant experiences. Thus, to broadly expand this definition, MSBand Diary, a mobile-based diary application, was developed not only containing the same function of handwritten diaries, but also providing different means to recognise and interpret past events by monitoring user’s physiological data obtained using wearable sensors. After a literature review focused on Affective Interaction, diary studies, physiological data in emotions and previous work in the topic, this paper guides through the whole process of application’s design and implementation. Different evaluation periods, where both quantitative and qualitative methods were applied, presented positive results about MSBand Diary being considered as a successful tool for self-reflection on past experiences, providing understandable graphical representations and a wide range of input modalities. Potential future works are also discussed in the end of this paper.

Keywords: diary application, mobile-based diary, physiological data, Microsoft Band, Affective Interaction, emotions, self-awareness, self-reflection
# Contents

Declaration 2  
Abstract 3  
Contents 4  
Introduction 7  
Literature review 9  
  1. Affective computing 9  
    1.1. How are emotions reflected in data? 10  
  2. Diary vision 11  
    2.1. Why mobile-based diaries? 11  
  3. Existing work 12  
    3.1. Affective Diary 12  
    3.2. Other mobile device-based diary studies 14  
    3.3. Commercially available diary mobile applications 15  
  4. Evaluation approaches 16  
  5. Conclusion 17  
Methodology 19  
Requirements Analysis 21  
  1. Initial requirements 21  
  2. Optional and emerging requirements 22  
Design 23  
  1. Devices overview 23  
  2. Android smartphone - Microsoft Band connection 24  
  3. System design 24  
  4. Interface design 25  
  5. Database design 30  
Implementation 33
1. Software Development Tools 33
2. Android fundamentals 33
   2.1. Android components 34
   2.2. Manifest file 35
   2.3. Resources 36
3. MSBand Diary’s first prototype 37
   3.1. Database support 37
   3.2. Activities 40
      3.2.1. MainActivity 40
      3.2.2. NewEntryActivity 44
      3.2.3. OneEntryActivity 44
      3.2.4. SendFeedbackActivity 45
   3.3. Broadcast receivers 46
      3.3.1. OnBootReceiver 46
      3.3.2. MSBandDataReceiver 48
      3.3.3. ResetDBReceiver 48
      3.3.4. ReminderReceiver 48
   3.4. Services 49
   3.5. Resources 50
4. MSBand Diary’s second prototype 51
   4.1. Database support 51
   4.2. Activities 52
      4.2.1. MainActivity 53
      4.2.2. NewEntryActivity 55
      4.2.3. CreateEmotionsActivity 56
      4.2.4. SettingsActivity 56
5. Testing and limitations 57
Evaluation 58
   1. Usability interviews (Evaluation Part 1) 59
      1.1. Material and methods 59
      1.2. Results 60
   2. Project evaluation (Evaluation Part 2) 64
      2.1. Material and methods 64
2.2. Results

2.2.1. Getting used to wearable sensors and a mobile-based diary

2.2.2. Input modes

2.2.3. Interpreting physiological data

2.2.4. Self-reflection

2.2.5. Conclusion on face-to-face interviews

2.2.6. Final usability survey

Future work

Conclusion

References

List of Figures

List of Tables

List of Snippets

List of Charts

Appendix A: UML Class Diagrams

Appendix B: Initial storyboards

Appendix C: Questionnaires and interview for project evaluation

Appendix D: Ethical approval and human consent

Appendix E: Risk Assessment
Introduction

A diary is commonly known as a book where people can keep a regular record of experiences and incidents with personal significance. In this way, it provides a useful resource to order thoughts and experiences in a daily basis, to be subsequently analysed by its own author. It could be simply defined as a way of expression, a tool to re-experience and record important, and often private, memories.

However, nowadays, diary’s definition is becoming more and more complex due to the use of smartphones. Media content is already easily accessible to everyone, creating text messages, videos or images on the move. Thus, this dissertation explores the possibilities of combining media content with data collected using sensor-based technologies, such as heart rate or skin conductance.

The possibility of using sensor data is explored because recent researches have matched emotions with a significant physical, bodily component. As main aim, this dissertation studies how influential events captured in terms of physiological correlates could be on the reflective aspects of keeping a diary, in conjunction with other ways of information capture such as media content or writing.

Consequently, the main objectives of this project include the following aspects:

- Develop a mobile application which enables users to express their everyday events in different ways.
- Create an efficient application in terms of user experience and performance, which motivates users to interact with the system.
- Evaluate qualitatively whether this kind of means can improve self-awareness, producing new reflections on past events by observing own physical reactions.

On a deeper level, regarding to the project’s main aim, data interpretation can only be developed by users in themselves. System can neither interpret this information nor suggest its meaning. Therefore, system should be limited to represent these data impelling users to
interact with the system, provide additional data and interpret the representation of their experiences.

As mentioned above, another objective, but no less important, was to create evolutionary prototypes of a mobile application which performs effectively and fulfils all the requirements for achieving a satisfactory user experience. It had to be able to retrieve data directly from the sensors generating an intuitive interface where the user can interact comfortably using their smartphones. Information provided by users was key in the application’s functionalities, therefore facilitating their inputs from any source is an essential requirement.

Microsoft Band 2 was the tool to monitor those physiological data needed in the application. It is a smart band with smartwatch features created and developed by Microsoft. MSBand 2 incorporates ten sensors, including heart rate monitor and Galvanic Skin Response sensors. Furthermore, the application was developed for Android devices.

In the following section, this paper will introduce a theoretical background in Affective Interaction, correlation between emotions and physiological correlates, and mobile-based studies, as well as several examples of past researches in common topics.
Literature review

Computing which deals with humans, humans who cope with emotions. This phrase could perfectly sum the field where this dissertation is present: Affective Computing. Consequently, this literature review is based on that area of expertise, as well as the application of diary studies in conjunction with mobile devices and wearable sensors. Furthermore, some existing works in Affective Computing field are critically analysed, highlighting the way their results were evaluated.

1. Affective computing

The original vision of affective computing is trying to recognise what users are experiencing when interacting with systems, their emotional responses to the interaction. Models are created and posteriorly adapted to modify this interaction by creating new emotional experiences or fixing problems that may be frustrating for the user. Affective computing is strongly connected with Artificial Intelligence, since it supports the possibility to understand users’ emotions. In its simplest form, affective computing is defined as: “Computing that relates to, arises from, or influences emotions” (Picard et al., 1997). These authors established the initial term of affective computing as the study and development of systems able to recognise and process aspects of human emotions.

However, equally with any movement within Human-Computer Interaction, there are different theoretical perspectives on the topic which adapt the original idea. An alternative branch to Picard’s cognitive model is Affective Interaction, based on the work of Boehner, DePaula, Dourish and Senger (Boehner et al., 2005). Their theory draws on phenomenology and understands emotion as constructed in interaction. Affective Interaction is concentrated in the critical view of whether somebody else’s emotion is possible to know by looking at faces, bodily gestures, or putting sensors to recognise physical reactions to emotions. In other words, it studies the possibility, or not, of reducing what users experience to a few measurable variables.
According to Kristina Hook (Hook, 2013), in Affective Interaction, the system does not think it knows what user’s emotion is by computing this emotion, but there is an interaction where emotion is constructed and interpreted in a dialog between the user and the system or between users through a system. In comparison with Affective Computing, Affective Interaction focuses on the emotions’n constructive process, where emotions take meanings between users in themselves, where the system is definitely not able to do this, as obviously do not know more about users than users in themselves.

Supporting Affective Interaction’s view, systems do not necessarily have to model or represent every single aspect of emotions and interaction does not need to be adapted to them, since they are not able to interpret the complexity of a variety of situations. For instance, users may become frustrated when systems interacts based on, for example, a smile: users do not have to be necessarily happy to smile, as they may be dealing with a tense situation which generates a nervous grin.

1.1. How are emotions reflected in data?

Research about emotions has demonstrated that autonomic responses can generate sensible data related with emotional reactions. For instance, as reported by Khalfa et al. (2002):

“Autonomic responses have been shown to vary according to reports of affective valence (i.e. pleasant or unpleasant) and arousal (i.e. calming or stimulating), which are two motivational determinants of emotions.”

Electrodermal activity can be measured providing autonomic data such as the Galvanic Skin Response. This physiological correlate, also known as Skin Conductance Response (SCR), is generated by rapid fluctuations in the eccrine sweat gland activity produced as a result of the liberation of acetylcholine by the sympathetic nervous system (Khalfa et al., 2002). This measurement is frequently used to measure humans’ arousal levels (Ståhl et al., 2009). Moreover, numerous studies have shown the SCR as a reliable measure of autonomic emotions’ expressions: in visual studies, when user looked at beautiful or emotional pictures (Lang et al., 1998) or in auditory modalities, by reproducing familiar sounds such as crying babies (Bradley et al., 2000). Unfortunately, a negative or positive valence of an arousal level can not be obtained only using bio-sensors. Nevertheless, the aim of this project is not to inform users about their emotions, but to help them to interpret this data about their
experiences. For example, they would know if a high level of arousal is produced in the gym, where humans generally sweat, and could omit this fact.

Heart rate also suffers significant variations by users living emotional experiences. As reported by Ekman and other researchers (Ekman et al., 1983), the heart’s activity changed not only between positive and negative emotions, but also distinguishing different negative emotions. Their research also indicates that heart rate increased more, for instance, in anger and fear, than in happiness.

2. Diary vision

The diary study is a method whose origin is partly found in both psychological and anthropological research (Gillham et al., 2005). It consists of a number of subjects recording data about their everyday lives for a certain amount of time. Obviously with the consent of those subjects, this data is gathered and analysed in different ways depending on its nature.

However, a number of potential concerns with such an approach have been highlighted by previous work (Grinter et al., 2007). Firstly, participants must often interrupt their current activity and manually make recordings. Furthermore, paper-based diary studies can be time-consuming, and the responsibility of writing all of their activities on paper could become repetitive. Therefore, this approach requires strong dedication from subjects during the evaluation period, as discussed by Xu Sun et al. (2013):

“Participants must be motivated to make a considerable effort to record their activities over the time-span of the diary, and it is easy for a participant to neglect to complete the diary.”

2.1. Why mobile-based diaries?

Over time, researchers have developed approaches which take advantage of mobile technologies to capture the necessary information. This solves a part of the limitation of paper-based diary methods mentioned in the previous section. As well as supporting by an electronic record of data, mobile-based diaries provide a number of advantages such as the ability to record audio, images and video, or passively collect data about subjects and their
There has been a number of examples that demonstrate the power of the mobile diary approach, and some of the characteristics of a successful implementation in mobile devices. Studies by Palen et al. (2002) reported diary studies with users using voice-mail paired with mobile and landline telephony to collect data. They informed that their diary provided highly valuable understanding, demonstrating that employing mobile phones in the collection of voice-mail data to study mobile activities and technologies can be especially powerful.

Other two studies by Ohmori et al. (2006) compared mobile phone-based data capture and conventional paper-based diaries in terms of data handling times for the researcher, the frequency of data entry and any ‘time lag’ between activity implementation and data entry. They found that the data handling time researchers need for data analysis was considerably reduced in the case of the mobile phone diary. Moreover, the frequency of data entry was higher for mobile phone diaries and the ‘time lag’ between activity implementation and data entry was shorter for the mobile phones compared to the conventional paper-based diary.

3. Existing work

This section will show previous research on the topic of this project, highlighting possible contribution that this dissertation may make to that field.

3.1. Affective Diary

Taking the conventional view of diaries, the Affective diary’s authors based their studies on how bodily experiences in everyday life were related to human’s emotions, forming what is called as “affective body memorabilia” (Lindström et al., 2006). The use of sensor-based technologies was also remarkably considered, as it enabled to record those physical experiences.

To study how these experiences may be used into the reflective process of a diary, Affective diary presented them to users through a system consisting of a mobile phone, body sensors and a tablet. Sensors had to be carried during the day to collect data about movement and
arousal levels, while mobile phone worked recording SMSs or photographs taken. At some point in the day, all this information had to be transmitted to the tablet via cable, where the application was set-up, to be processed. The data were shown in shaped, coloured figures, created from sensor’s data, over a timeline. Users were also invited to add notes on this process to reflect their dairy experiences.

Figure 1. Affective diary: adding a note (extracted from Lindström et al., 2006)

Figure 2. Affective diary: adding a picture (extracted from Lindström et al., 2006)
Bodily postures’ representations were discussed and tested with users, and represented using data provided by sensors. Figures showed different body positions, while colours were directly influenced by the arousal level users experimented over time. Such levels were measured by Galvanic Skin Response sensor placed on users’ upper arms.

In order to evaluate the system, four participants were asked for testing the Affective Diary for several weeks divided in two phases. After each phase, users were interviewed to show their diaries and discuss about how positive the application was for their interpretations. As a result, researchers received videos of interviews and subjects’ complete diaries that were posteriorly analysed in terms of users’ satisfaction, self-reflection and self-learning.

As an outcome of evaluation, Affective diary showed participants saw and handled the materials provided in different ways, both to find subjectives experiences such as their spirituality and to simply remember past events.

This diary included several of the functionalities we would like to explore in this project. However, its research is focused on corporal gestures and body positions, rather than physiological correlates. Users had to use a certain kind of sensors to enable the diary to work properly, and probably due to the technology used, it could not be use over time. For example, due to Affective Diary was a prototype, users could not upload sensors’ data by themselves, so recordings were uploaded by the research team several days before each interview.

3.2. Other mobile device-based diary studies

Numerous studies have also demonstrated the power of mobile diary approaches in participants’ experiences. Among others, a research which investigated users’ perception of serendipity (Sun et al., 2013) deserves to be highlighted in this literature review. There are several theories which link serendipity with end-user perception. Therefore, the purpose of this study was applied to get users’ recordings in their everyday life, and posteriorly, suggest them to discuss about serendipity using this captured information.

Consequently, an Android application was developed for this purpose. The system allowed users to read and write their own diary by including audio, video, text or photograph in those moments when users may experience serendipity cases.
Eleven participants from different backgrounds tested the application during a period of seven days, to be later interviewed using their diaries as a resource. These interviews were qualitative in nature and were based on participants’ experiences using the application.

As a result of this evaluation, a set of data containing diaries and several hours of interviews were obtained. Evaluation was successful, as this data provided a good understanding of how users experience serendipity in different contexts.

3.3. Commercially available diary mobile applications

At present, a wide range of applications designed for mobile devices can be found in the most popular “app markets”. However, most of these are based on user inputs, not taking into account other functionalities that modern devices could provide, such as geolocation. Another noticeable aspect is that none of them utilise additional sensors in their functioning, depending totally on information provided daily by the user.
4. Evaluation approaches

Once evaluation techniques in existing works have been slightly described, evaluation approaches need to be discussed, including potential issues that will be also critically analysed.

As seen in the last section of this literature review, there are several researches aiming to know how successful diary studies are in evaluation approaches. For instance, both Affective Diary and Serendipity research tried to incorporate the system as part of users’ daily lives. In that way, users totally controlled the system by themselves, interacting and utilising any incorporated functionality. These are clear examples of evaluation “in the wild” (Crabtree et al., 2013), which differ from typical approaches such as usability evaluation in labs. Researchers are increasingly supporting this kind of evaluation: studies carried out in-situ obtaining users’ experiences on the go, highlighting the importance of building technology adapted to everyday life.

The most important part in a “wild” design is to develop and evaluate prototypes of the whole system to be posteriorly tested by participants. Data is collected by contemplating and recording people’s experiences and how their reflections change over time using system’s prototypes. For that reason, studies’ duration increases its relevance over sample size. The number of participants would be no longer too important, as noticed in Affective Diary and Serendipity research, where four and eleven users took part in the study respectively.

This characteristic, among others that will be mentioned hereafter, places evaluation “in the wild” into the category of qualitative evaluation methods. In terms of data collected, while quantitative methods basically gather numbers, qualitative ones consider words, videos, images or audios, since these kinds of information help to understand why the system has been developed and how it influences over users, rather than providing hard and scientific information.

Several techniques are used to collect and study qualitative data. As seen in Existing work section, Affective Diary’s researchers used interviews in two different phases during the evaluation period, as well as obtaining users’ recordings. With respect to Xu Sun et al.’s research, the same techniques were used, but they included additional types of information, such as videos, audios and photographs.
However, as in any evaluation method, several limitations and issues can be encountered. Involving a smaller number of participants also leads to the impossibility of generalising the research’s findings. Moreover, the data collection is more time-consuming than in quantitative methods, although this issue is counteracted using a shorter sample size.

The self-sufficiency of users evaluating the system also causes problems in their implication. Participants may not remember to interact with the system daily, as happened in the Serendipity research. A possible solution to avoid this issue could be to add non intrusive reminders in the application, which would be triggered at certain points of the day selected by the user.

Finally, and most importantly, the effect that the system can provoke on the user during the evaluation period may not be always positive, as occurred in Affective Diary. Diary applications make participants to reflect about their daily events, and sometimes, they can become conscious about problems they did not want to know. System in itself can not avoid this issue, but in this research, the application will try to find the right balance between not suggesting what should be interpreted in each case and motivating users to participate in the interaction.

These methods, techniques and issues described above motivated the evaluation plan for this dissertation, which is found in the section Evaluation of this paper.

5. Conclusion

In this literature review, we explored the Affective Computing field, and more specifically Affective Interaction principles, where this dissertation belongs to. This focused our objective on developing an application that does not interpret users’ emotions, but also models an environment where users were able to identify them while interacting with the system.

Subsequently, the system should provide several tools which help this process occur. Relying on the capabilities of wearable sensors (Microsoft Band), we found the possibility of building graphical representations of physiological data (heart rate and GSR) due to how emotions are reflected on them, where subjects could interpret past events.
Moreover, a diary approach was also found appropriate to model this environment considering all its risks, but also its advantages in qualitative evaluation methods. It requires strong dedication from subjects, but building the system in a mobile device would mitigate that risk considerably, as subjects could use its personal smart phone and interact with the system on-the-go.

Existing diary applications were studied as well, some commercially available and others belonging to private researches. We could extract knowledge about how these applications were designed and how people used them, as well as past researches’ evaluation methods, where qualitative evaluation techniques and evaluation “in the wild” stood out over others. From this review, no available software for Android combining all these approaches was found.
Methodology

In software development, before making decisions about what methodology to use, certain aspects such as risks, costs, requirements, or deadlines must be considered. In particular, our software application may include numerous functionalities, or need to modify initial requirements depending on users’ evaluation. This is why we followed a Software Prototyping model and, more specifically, Evolutionary Prototyping.

In comparison with other more rigid methodologies such as Waterfall model, where each process phase must be totally specified before the next phase can begin, Prototyping can improve the quality of functionalities and requirements due to the use of user-centered design. It also avoids exponential costs by detecting user requirements early in development, as Prototyping requires constant interaction between users and system. For this reason, this methodology produces more satisfactory software in terms of performance and usability.

However, there were risks involved in this methodology. As Prototyping does not often use a complete vision of the project, it can lead to incomplete specifications, not including important software engineering requirements and creating products difficult to maintain. Wrong time management may become another disadvantage of Prototyping. In spite of these disadvantages, and since this project relies heavily on users’ feedback, these possible risks are assumed in the software development process and Prototyping is considered as the most appropriate choice for this project.

Generally, a prototype only includes a few aspects of the final software. However, it presents the following benefits:

- Valuable feedback from users can be received early in the project.
- Confusing or unnecessary functionalities can be identified and discarded easily.
- It also allows developers to get a better understanding of the whole project, producing more efficient approaches and sometimes decreasing the software development’s time.
Evolutionary Prototyping consists of the implementation of a solid prototype which will work as a basis for the rest. Further versions improve this first prototype, including new requirements or features that could not be defined in the initial design. Even though they are not complete versions, first prototypes are totally able to be used until further versions release. Considering that, Prototyping can be divided in several steps:

1. Basic requirements’ identification: Initial requirements must be defined, focusing on users’ inputs and results.

2. Initial Prototype’s development: using horizontal development based on user interaction rather than complex functionalities.

3. Users’ evaluation: the prototype is tested generating feedback on future improvements.

4. Revision and enhancement: this stage could be considered as the first one of the next prototype. Using the last evaluation’s results, another prototype starts to be created containing new functionalities and requirements.

The application’s first prototype did not contain all the planned functionalities, but users were already able to control their physiological data represented on graphs and provide simple ways of inputs, such as text or icons. Afterwards, its performance and usability were evaluated, also identifying errors and issues arises from users’ feedback. All this information was used to develop the next prototype of the application’s development process.
Requirements Analysis

As detailed by Prototyping principles, the first prototype should not contain all the planned functionalities, since it is used to detect new requirements, previously unknown, which help to improve the first version of the application, making it more beneficial from users’ point of view. Therefore, only initial requirements needed to be identified for the first prototype, as well as others optional likely to be included in future versions. During the evaluation periods, new requirements emerged from users feedback to create different system’s functionalities.

1. Initial requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Data retrieval</td>
<td>Both MSBand and Android mobile device are connected to each other via Bluetooth, in order to make the app retrieve the information provided by the MSBand, including GPS data, heart rate and Galvanic Skin Response. This information shall be accessible and read properly to be stored and involve changes in the application.</td>
</tr>
<tr>
<td>2. Graphs creation</td>
<td>The data collected will be shown in a plain graph, which will be created automatically over days.</td>
</tr>
<tr>
<td>3. Simple user input</td>
<td>Application must allow the user to include text or icons at some point in the day, providing extra information to identify a high level of arousal in the graphs.</td>
</tr>
<tr>
<td>4. Intuitive interface</td>
<td>An intuitive user interface will be incorporated so that users do not find difficulty interacting with the application.</td>
</tr>
</tbody>
</table>

Table 1. Initial requirements
2. Optional and emerging requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5. More complex types of user input</td>
<td>Application would allow the user to include some types of files such as pictures, audio clips or emotions’ names to provide extra information to the application.</td>
</tr>
<tr>
<td>6. New graphs incorporated</td>
<td>The data collected would be shown in new types of graph, which would be created automatically over days.</td>
</tr>
<tr>
<td>7. Data interpretation</td>
<td>By analysing the data, the application would provide precise summaries of arousal levels, identifying, for instance, possible problems at work or busy lifestyle.</td>
</tr>
<tr>
<td>8. Location accuracy</td>
<td>By providing addresses, the application would identify usual places frequently visited by the user.</td>
</tr>
<tr>
<td>9. Incorporating in the MSBand</td>
<td>By a new tile in the MSBand, users would have access to a simple interface where app notifications can be read and some simple information provided to the application.</td>
</tr>
</tbody>
</table>

Table 2. Optional requirements

As can be seen later in this paper (see point 3 in Implementation section and 1.2 in Evaluation) in further sections, all the initial requirements were successfully fulfilled, except for the GPS integration, as it was not valued enough in early evaluation questionnaires.

Regarding the optional requirements, some of them were finally discarded because of time restrictions or an unfavourable feedback from users (requirements 8 and 9 in Table 2). However, new requirements were obtained during the project’s development and different evaluation periods, which will be introduced in Table 3 and explained in Design and Implementation sections.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10. Sending of user feedback</td>
<td>Application would allow the user to send immediate feedback about any aspect of the application. This feedback would be used in further prototypes or debugging.</td>
</tr>
<tr>
<td>11. Notification system</td>
<td>User should be able to enable daily notification which will work as reminders.</td>
</tr>
</tbody>
</table>

Table 3. Requirements obtained during evaluation period
Design

Every software contains a design process where solutions are provided to fulfil a set of requirements, including different techniques and tools used to achieve those objectives. This section includes decisions on systems functionalities, user interfaces and database, together with the changes these parts suffered during the whole project.

1. Devices overview

In terms of mobile operating systems, we found Android as the best selection, since it provides the following advantages:

• Portability: Android apps are developed using Java programming, and could easily be adapted to different operating systems like Symbian, Ubuntu or Chrome OS.

• Java: Android uses Java for application development, a powerful, frequently used language utilised on multiple devices and operating systems.

• Market share: according to IDC (2015), Android dominated the number of smartphones shipped in 2015. In these terms, our application will be widely available for users.

Therefore, an Android smartphone shall be necessary to set up the MSBand Diary. The Android version had to be 4.2 or higher due to reasons explained in the next point.

Regarding to complementary devices, wearable sensors have been used in research to capture information about subjects in different situations, as well as allowing the collection of physiological data on-the-go about peoples’ activity. For that reason, Microsoft Band 2 has been selected in this project to monitor users’ data required to carry out the development of the application. For this project, what is really needed from Microsoft Band 2 is the data collected by its optical heart rate sensor and Galvanic skin response sensor, which measures conductivity of skin and provides a solution to determine the arousal level of the subjects. Microsoft Band 2 is also easily connectable via Bluetooth with most smart phone devices.
2. Android smartphone – Microsoft Band connection

In order to connect MSBand Diary application to the Microsoft Band sensors, we made use of the Microsoft Band SDK. This software development kit was designed to enable the communication with the Microsoft Band sensors from any application. To that end, the application needs to subscribe to the band’s sensors, whose data are sent via Bluetooth. However, to successfully connect with the Microsoft Band, its SDK relies on the Microsoft Health App (publicly available). Users would need to set up this application to pair the Microsoft Band with their Android devices before using MSBand Diary application. The Figure 7 shows the different devices taking part in the system, as well as possible inputs received from users.

The minimum-supported Android API version for the SDK is 17 (Android 4.2). It also requires that the application receives permission to use the Android device’s Bluetooth, as this is the method to connect to the Microsoft Band.

3. System design

In this section, the application’s initial design concerning the main functionality is described briefly. It consists of the core operations to perform the basic requirements specified in previous sections.
1) At a very early stage, when the application is launched, the system checks if there is a Microsoft Band paired with the Android phone. If so, the user is required to accept a consent allowing MSBand Diary to subscribe to the Band’s sensors.

2) At that same stage, the application also creates two Android receivers (a component that responds to system-wide broadcast announcements) setting a daily reminder notification and a database reset at the end of every day (see point 5, Database design).

3) If the consent is granted, the application starts receiving physiological data (heart rate and Galvanic skin response) that will be stored in its corresponding database. A service running in the background plays the role of data receiver and is also responsible for updating the database.

4) In the meantime, the application displays a chart in the foreground using the Microsoft Band sensors’ data. The chart represents heart rate and arousal level (based on Galvanic Skin response) during the current day divided in periods of 15 minutes.

5) At the time, users have the ability to create diary entries by selecting a spot in the chart or adding it manually, navigate through the application to find previous entries or visualise other charts which generates weekly and monthly summaries.

4. Interface design

As for user interface, it was necessary to generate an early version which fulfilled the initial requirements and enabled users to interact with the system from its first prototype. To obtain an idea about the user interface’s appearance, different storyboards were designed before developing the application’s first prototype.

As can be seen in the first storyboard (see Appendix B), the application had to retrieve data properly (see Requirement 1) to posteriorly generate a graph where that information would be reflected (see Requirement 2). At that point, graph showed variations in heart rate and arousal level. User would be also able to create entries (see Requirement 3) by text or simple icons. Based on this model, a first prototype’s user interface was developed (see Figure 8).
Figure 8. First prototype’s user interface
Unlike the first storyboard, the application does not finally contain a login system. The phone unlocking was considered enough as a security mechanism. The “Settings” interface was not also necessary at this stage, as it was supposed to control the connection with the Microsoft Band and, as already briefly mentioned, this process is performed by the Microsoft Health App. However, a “Send feedback” button was provided to enable users to email the developer indicating errors or suggestions for further prototypes (see Requirement 10). The information collected by using the method was also part of the project evaluation since it provided useful feedback from an early stage.

After getting feedback from first usability evaluation questionnaires and in order to provide new functionalities initially planned, the user interface changed considerably in the second prototype:

- At this point, users can include pictures in their entries by choosing them from the gallery or taking new ones using the phone camera. The applications also provides an editable list of emotions to help users describe past experience in a more interactive way, minimising the quantity of text input.
- Entries can now be filtered by week, so “Entries list” interface does not show the whole set of user entries.
- As planned initially, users are able to explore the media content included in their entries in a different screen, as well as obtaining weekly and monthly summaries of their entries based on emotions and “positivity” icons found in an amount of time provided by themselves. Monthly summaries were mainly implemented for future use, as the evaluation period for this project is less than one month.
- This last prototype does include a Settings screen, where users can set a daily notification which reminds to write diary entries at a selected time. Apart from the Settings choice, a small menu also contains an “About” section which built a pop-up screen showing details about the application as a part of a more extended project and explaining how the charts are built.

All these changes can be appreciated in the Figure 9, which was built based on the last prototype’s user interface.
Figure 9. Second prototype’s user interface
Most of interfaces are designed to support vertical screen orientation (portrait). However, in this last prototype, the “New entry” interface also enables horizontal orientation to facilitate note-taking, as virtual keyboard becomes larger.

Figure 10. Providing horizontal orientation while creating entry

The whole user interface design was developed following the Android Material Design principles (Google Material website, 2014). Other possible improvements and new functionalities obtained from different feedbacks not included in this project due to time restrictions will be specified for further application versions in Future Work section.

5. Database design

As already mentioned, the application works with different types of information which obviously need to be stored in databases. Dealing mainly with numeric data from users’ physiological correlates and text provided by users, our application uses the SQLite Database for Android. SQLite is a relational database management system contained in a C programming library. In contrast to many other database management systems, SQLite is not a client–server database engine. Rather, it is embedded into the end program. As other RDMS, it is ideal for repeating or structured data.

Consequently, this data will be stored locally in users’ mobile devices, thereby reducing the risk of inappropriate reading of personal information. Moreover, any SQLite databases created in Android will be accessible by name to any class in the application, but not outside the application.

The database development was performed at the first stage of the application design. Initially, the database contained two different tables storing diary entries data and user’s physiological correlates (see Figure 11).
Initially, diary entries consisted of a title, a date (in yyyy-MM-dd format), a time (in HH:MM format), an icon corresponding to a integer value (depending on the icon the user selects to identify positiveness of the event) and the text content of the entry.

As for the physiological correlates storage, the table contained the data for the current day. A diary is supposed to be completed day by day. Therefore, storing user’s heart rate and GSR of previous days would not be necessary. As already mentioned, an application service reads sensors’ data in the background. It is triggered every 30 seconds, an amount of time frequent enough to subsequently create accurate charts for this data. In early tests, this period was reduced to 5 seconds, but it led to a Microsoft Band’s poor battery performance. If days are divided in period of 30 seconds, each day, and therefore the msbanddata table, would contain 2880 parts (or rows in a table). For that reason, together with the table creation, 2880 rows are added with null values in heart rates and GSR, but containing times from 00:00:00 to 23:59:30 in 30-second periods. Thus, the MSBand Diary service read sensors’ data and store them in the row containing the closest time to the reading moment. It is important to note that if due to any reason a service is not running and rows are not updated, those data values are not used in chart creation.

However, after receiving feedback from users about the first application prototype, the database design changed to allow new content storage (see Figure 12).

The new version allowed users to include different emotions in their diary entries by using a list. The list of emotions can be edited by users, adding or deleting them from the table emotions. In this way, entries table was updated to contain several emotions’ names from that new table. The last prototype also allows users to add pictures in entries, so pictures’ complete path are registered in a new column called image in the entries table. The way this path is build will be discussed in the Implementation section.

msbanddata table did not suffer any change at this stage.
Figure 12. Last prototype’s Entity - Relationship model
Implementation

This section will analyse the whole process of the application’s implementation, as well as a description of all the tools utilised and a briefly explanation about the Android platform. As already explained, the development of the application was divided in two different prototypes, so the application’s functionalities will be addressed separately in two subsections, including any problem faced during each stage.

Complete source code of some of the main classes implemented can be found in the ZIP file attached to this paper, and different class diagrams showing relationships and dependencies between classes in the Appendix A section of this paper.

1. Software Development Tools

Android Studio was the software development tool used to build the application. Official IDE for Android, it is freely available under the Apache License 2.0. Because of the application is required to generate graphical representations of the user’s physiological correlates, HelloCharts for Android (Leszek Wach's Github page, 2011) was utilised to create different charts. It is a charting library for Android compatible with API 8+ (Android 2.2) also available under Apache License 2.0. GIMP (GNU Image Manipulation Program) was the graphic editor used to create images and icons for the application’s user interface.

2. Android fundamentals

Java is the programming language used in Android. Java code together with other application resources are compiled into an Android package (.apk file), containing all the necessary content to be installed in an Android device. Once installed, each application works in its own Linux process, being controlled by Android which applies the principle of least privilege. Therefore, Android starts and ceases the application’s process when necessary.
Android also provides a secure environment, where it requires special permissions to every app to gain access to different Android components.

2.1. Android components

As explained in the Android API guide (Android Developers website, 2016), every Android application consists of a set of Android components working together. Each component has its own functions and lifecycle, so they must be treated differently.

- Activities: implementations of Activity class, they build a single screen within a user interface. MSBand Diary contains 7 different activities used to, for example, create a new entry in the diary. Being independent of each other, the set of activities work together to create the user interface. To reuse code, activities use fragments at times, which represent user interface's sections performing a concrete action.

- Services: running in the background, services are responsible for executing long-running operations or performing, for example, network processes. Independent of the user interface, MSBand Diary implements a service responsible for retrieving data from the Microsoft Band. A service is implemented as a subclass of Service.

- Broadcast receivers: Android system implements a mechanism of announcements, such as one alerting of a complete system reboot, which need to be listened by means of broadcast receivers. Broadcast receivers can also catch announcements generated during the own functioning of the application. They are a critical part of MSBand Diary application, since they act as a bridge between different components. Usually not containing a large amount of work, they are responsible for triggering repetitive actions, such as creating a daily notification at a given time or resetting the database table containing physiological correlates every day. A broadcast receiver is implemented as a subclass of BroadcastReceiver and delivered as an Intent object.

Content Providers are the remaining component of Android, but they have not been used during the implementation of MSBand Diary.

All these components need to be connected and activated at runtime. That is the role which Intents play. By defining a message, they carry out an action to activate, for example, a delivery or display of data in cases of activities and services.
2.2. Manifest file

Every Android application contains a Manifest file where they register all their components. This file is accessed before the Android system starts any application’s component. It consists of a XML file located at the root application’s directory.

The manifest also identifies any permission that the app requires. In our case, MSBand Diary needs permission to Bluetooth access, read/write access to the external memory, receiving a notification when the device boots or read-access to the Microsoft Band sensors. Moreover, the Manifest file declares other application features such as the minimum API required, hardware features, necessary API libraries or application’s name and launcher icons.

For all these features, the Android Manifest file provides a clear overview of all the components present in an application. Snippet 1 shows the code for MSBand Diary’s Manifest file.

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="lg21.macs.hw.ac.uk.msbanddiary">
  <uses-permission android:name="android.permission.BLUETOOTH" />
  <uses-permission android:name="com.microsoft.band.service.access.BIND_BAND_SERVICE" />
  <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
  <uses-permission android:name="android.permission.WAKE_LOCK" />
  <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

  <application
    android:name=".MSBandDiaryApplication"
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity
      android:name=".MainActivity"
      android:label="@string/app_name"
      android:screenOrientation="portrait"
      android:theme="@style/AppTheme.NoActionBar">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
        <category android:name="android.intent.category.NOTIFICATION_PREFERENCES" />
      </intent-filter>
    </activity>
    <activity
      android:name=".NewEntryActivity"
      android:label="@string/title_activity_new_entry"
      android:theme="@style/AppTheme.NoActionBar" />
    <activity
      android:name=".SettingsActivity"
      android:label="@string/title_activity_settings"
      android:screenOrientation="portrait" />
  </application>
</manifest>
```
Apart from its source code, an Android application uses a set of resources to represent its visual appearance. For example, MSBand Diary includes a set of layouts, menus, styles and colours defined with XML files.

2.3. Resources

Snippet 1. Manifest file

```xml
<activity
    android:name=".OneEntryActivity"
    android:label="@string/title_activity_one_entry"
    android:screenOrientation="portrait"
    android:theme="@style/AppTheme.NoActionBar" />
<receiver
    android:name=".MSBandDataReceiver"
    android:enabled="true"
    android:exported="false">
</receiver>
<receiver
    android:name=".OnBootReceiver"
    android:enabled="true"
    android:exported="false">
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED" />
    </intent-filter>
</receiver>
<receiver
    android:name=".ReminderReceiver"
    android:enabled="true"
    android:exported="false">
</receiver>
<receiver
    android:name=".ResetDBReceiver"
    android:enabled="true"
    android:exported="false">
</receiver>
<service
    android:name=".MSBandDiaryService"
    android:enabled="true"
    android:exported="true" />
<activity
    android:name=".SendFeedbackActivity"
    android:label="@string/title_activity_send_feedback"
    android:parentActivityName=".MainActivity"
    android:theme="@style/AppTheme.NoActionBar">
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value="lg21.macs.hw.ac.uk.msbanddiary.MainActivity" />
</activity>
<activity
    android:name=".CreateEmotionsActivity"
    android:label="@string/title_activity_create_emotions"
    android:parentActivityName=".NewEntryActivity"
    android:theme="@style/AppTheme.NoActionBar">
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value=".NewEntryActivity" />
</activity>
</application>
</manifest>

36
Every resource is identified by the application by means of an unique ID, which is used to reference the resource from the application’s code. For instance, Figure 13 shows the launcher icon (named `ic_launcher.png`) built for MSBand Diary application, which generated a resource ID named `R.mipmap.ic_launcher` (because the icon was stored in `mipmap` folder), which can be used to reference the image and insert it where necessary.

![Figure 13. MSBand Diary launcher icon](image)

3. **MSBand Diary’s first prototype**

At this stage, MSBand Diary was supposed to fulfill the initial requirements of the project consisting of Microsoft Band data retrieval, the chart’s creation based on those data and support for creating and listing diary entries using simple input methods. The visual results after building the first prototype can be observed in the previous section, but this point explains what Android components made it possible to obtain those results.

### 3.1. Database support

As can be seen in the Design section, the application needed to store data locally in a persistent way. For this purpose, it was necessary to provide support to work with SQLite databases. We made use of the Android’s SQLite implementation, which uses a private storage space where only the application has access.

Related to database support, a class was implement to manage all the operations involving the database, such as creation and deletion of tables or insertion and retrieval of data. This class extends `SQLiteOpenHelper`, a class provided by Android to manage local databases. Our database helper class creates two different tables in the database: “entries” storing data about diary entries and “msbanddata” containing the user’s physiological data daily (see Snippet 2).
package lg21.macs.hw.ac.uk.msbanddiary;

import ...  

public class MSBandDiaryDBHelper extends SQLiteOpenHelper{

    private static final String TAG = "MSBandDiaryDBHelper";

    // Private class instance for Singleton instance
    private static MSBandDiaryDBHelper sInstance;

    public static final String DATABASE_NAME = "MSBDiary.db";
    private static final int DATABASE_VERSION = 1;
    public static final String ENTRY_TABLE_NAME = "entries";
    public static final String ENTRY_COLUMN_ID = "_id";
    public static final String ENTRY_COLUMN_TITLE = "title";
    public static final String ENTRY_COLUMN_DATE = "date";
    public static final String ENTRY_COLUMN_TIME = "time";
    public static final String ENTRY_COLUMN_ICON = "icon";
    public static final String ENTRY_COLUMN_EMOTION = "emotions";
    public static final String ENTRY_COLUMN_NOTES = "notes";
    public static final String ENTRY_COLUMN_IMAGE = "image";

    public static final String BAND_DATA_TABLE_NAME = "msbanddata";
    public static final String BAND_DATA_COLUMN_ID = "_id";
    public static final String BAND_DATA_COLUMN_TIME = "time";
    public static final String BAND_DATA_COLUMN_HEART_RATE = "heart_rate";
    public static final String BAND_DATA_COLUMN_GSR = "gsr";

    public static synchronized MSBandDiaryDBHelper getInstance(Context context) {
        if (sInstance == null) {
            sInstance = new MSBandDiaryDBHelper(context.getApplicationContext());
        }
        return sInstance;
    }

    private MSBandDiaryDBHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE " + ENTRY_TABLE_NAME + "(" +
                ENTRY_COLUMN_ID + " INTEGER PRIMARY KEY, " +
                ENTRY_COLUMN_TITLE + " TEXT, " +
                ENTRY_COLUMN_DATE + " TEXT, " +
                ENTRY_COLUMN_TIME + " TEXT, " +
                ENTRY_COLUMN_ICON + " INTEGER, " +
                ENTRY_COLUMN_NOTES + " TEXT, " +
        ");

        db.execSQL("CREATE TABLE " + BAND_DATA_TABLE_NAME + "(" +
                BAND_DATA_COLUMN_ID + " INTEGER PRIMARY KEY, " +
                BAND_DATA_COLUMN_TIME + " TEXT, " +
                BAND_DATA_COLUMN_HEART_RATE + " INTEGER, " +
                BAND_DATA_COLUMN_GSR + " INTEGER")");

        initialiseMSBandTable(db);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + ENTRY_TABLE_NAME);
        db.execSQL("DROP TABLE IF EXISTS " + BAND_DATA_TABLE_NAME);
        onCreate(db);
    }

    // Other methods to create, delete, get and edit data from database ...
}
MSBand Diary’s SQLite database is used across the entire application within services, applications and fragments. To ensure that only an instance of SQLite database is accessed simultaneously, the Singleton pattern was applied. As showed in Snippet 2, our helper class contains a private instance of itself which can only be accessed using the synchronised getInstance() method. If this instance has not been initialised yet, this method will do it. If one already exists, getInstance() will simply returned it.

According to the approach to store user’s heart rate and Galvanic Skin Response data described in the Design section, once “msbanddata” table is created, the method initialiseMSBandTable (see Snippet 3) adds 2880 rows with null values in heart rates and GSR, but containing times from 00:00:00 to 23:59:30 in 30-second periods.

```java
public void initialiseMSBandTable(SQLiteDatabase db){
    ContentValues contentValues;
    for(int i=0;i<2880;i++){
        try {
            Calendar calendar = Calendar.getInstance();
            calendar.set(Calendar.HOUR_OF_DAY, 0);
            calendar.set(Calendar.MINUTE, 0);
            calendar.set(Calendar.MILLISECOND, 0);
            calendar.set(Calendar.SECOND, i * 30);
            contentValues = new ContentValues();
            contentValues.put(BAND_DATA_COLUMN_TIME,
                                new SimpleDateFormat("HH:mm:ss").format(calendar.getTime()));
            contentValues.put(BAND_DATA_COLUMN_HEART_RATE, 0);
            contentValues.put(BAND_DATA_COLUMN_GSR, 0);
            db.insert(BAND_DATA_TABLE_NAME, null, contentValues);
            db.setTransactionSuccessful();
        } catch (Exception e) {
            Log.d(TAG, "Error while trying to create MSBand table");
        } finally {
            db.endTransaction();
        }
    }
}
```

Snippet 3. MSBand table initialisation

MSBandDataDBHelper also contains other methods called from different activities to display, create, update or delete data from the database. For example, the main interface has a list of the last five diary entries created. To perform this operation, it is necessary to get a Cursor object to the result set returned by a database query (see Snippet 4).

```java
public Cursor getLastEntries() {
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor res = db.rawQuery("SELECT " + ENTRY_COLUMN_ID + ", " +
                           ENTRY_COLUMN_TITLE + ", " + ENTRY_COLUMN_DATE + ", " +
                           ENTRY_COLUMN_TIME + ", " + ENTRY_COLUMN_ICON + ", " +
                           ENTRY_COLUMN_NOTES + " FROM " + ENTRY_TABLE_NAME + " ORDER BY date(" +
                           ENTRY_COLUMN_DATE + ") DESC, time(" + ENTRY_COLUMN_TIME + ") DESC Limit 5",
                           null);
    return res;
}
```

Snippet 4. Getting last 5 diary entries
3.2. Activities

Four different activities were included in the application at this stage (see Table 4). We can understand an activity as a single screen in a user interface. So as not to extend this paper too much, only essential parts of the source code will be included.

<table>
<thead>
<tr>
<th>Activity name</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>MainActivity</td>
<td>Launcher activity which use fragments to show different screens.</td>
</tr>
<tr>
<td>NewEntryActivity</td>
<td>Activity which generates a UI screen to create a new diary entry providing several inputs fields. It is also used to edit past entries.</td>
</tr>
<tr>
<td>OneEntryActivity</td>
<td>It displays all the information stored about a diary entry.</td>
</tr>
<tr>
<td>SendFeedbackActivity</td>
<td>It provides a screen which allowed to get immediate feedback from subjects by sending emails.</td>
</tr>
</tbody>
</table>

Table 4. First prototype’s activities

3.2.1. MainActivity

This activity consists of the main screen of the application, the activity which works as launcher when the user opens the application. It also allows to navigate to other screens. Responsible for providing a Navigation Drawer, a floating button to add diary entries, and a toolbar containing the “Feedback” button, *MainActivity*’s role is to adapt the content of the user interface depending on the user’s selection on the Navigation Drawer (see Figure 14). For this purpose, it uses different fragments for each choice.

![Figure 14. First prototype’s MainActivity screen](image)
**HomeFragment**

This is the fragment used by default. Like all fragments, it uses a XML layout file where all the UI components are declared to be later referenced from its code. For example, the Snippet 5 contains an example of a list declaration which will be subsequently populated from Java code. Visually, this fragment displays the main chart where users can visualise their heart rate and arousal level data, as well as a list containing the last five diary entries.

```xml
<ListView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/listView_Entries"
    android:divider="@android:color/transparent"
    android:dividerHeight="4.0sp"
    android:choiceMode="singleChoice" />
```

Snippet 5. UI component declaration

From a programming view, **HomeFragment** class has several responsibilities:

1) It checks if the Microsoft Band is connected using an asynchronous task. If so, in its first launch, it shows a dialog asking the user to accept a consent to access the Heart Rate sensor of the Microsoft Band. It is a Microsoft Band SDK's requirement. If Microsoft Band is not paired, it displays a pop-up message. Snippet 6 shows the method which checks if Microsoft Band is paired with the Android device.

```java
private boolean getConnectedBandClient() throws InterruptedException, BandException {
    // client is an instance of BandClient
    if (client == null) {
        BandInfo[] devices = BandClientManager.getInstance().getPairedBands();
        if (devices.length == 0) {
            return false;
        }
        client = BandClientManager.getInstance().create(getActivity().getBaseContext(),
                                  devices[0]);
    } else if (ConnectionState.CONNECTED == client.getConnectionState()) {
        return true;
    }
    return ConnectionState.CONNECTED == client.connect().await();
}
```

Snippet 6. Checking Microsoft Band connection

2) **HomeFragment** retrieves data from the database, as it needs to build the chart based on the user’s physiological data and a list of entries. In order to do this, it queries the database using the database helper class, obtaining two different Cursor objects which will be later adapted to UI components. Snippet 7 shows how to access the database and obtain the desired data using methods implemented in our **MSBandDiaryDBHelper**.
3) Once *HomeFragment* obtains data relating to heart rate and GSR, it needs to adapt them to create the chart. First of all, the application considers heart rate range between 30 to 180 beats a minute according to American Heart Association website (2016). Galvanic Skin Response data are specified in kOhms by Microsoft Band sensor. As this measure varies considerably from person to person, we needed to enable a wide GSR range. Application considers 25000 kOhms as a maximum GSR (Vijaya et al., 2013). While most of people knows how to interpret heart rate data, we needed to design a visually understandable approach for users to represent GSR data in a chart. As mentioned in Literature Review section, GSR decreases to indicate a low arousal level and increases in stressful or exciting situations. Therefore, considering that both Heart Rate and GSR data are represented in the same chart axis, the application adapts GSR values to Heart Rate range, finally showing an intuitive “Arousal level” zone instead of GSR values. Heart Rate and formatted GSR values are finally displayed in the main chart using tools provided by HelloCharts library.

The layout resource together with the data retrieved programatically generate the UI section showed in Figure 15. It should be mentioned that the “+” button belongs to *MainActivity* and is shared by all the fragments.

```java
// dbHelper is an instance of MSBandDiaryDBHelper
dbHelper = MSBandDiaryDBHelper.getInstance(getContext);
final Cursor cursor = dbHelper.getLastEntries();
final Cursor curData = dbHelper.getAllBandData();
```

Snippet 7. Getting database data

![Figure 15. First prototype's HomeFragment screen](image)
EntriesFragment

It is the fragment used when the user selects “Entries list” in the navigation drawer.

1) At this first prototype, this fragment displays all the entries found in the database in a custom list sorted by date. The order is established directly using SQL code in the corresponding `MSBandDiaryDBHelper` class method (see Snippet 8).

```java
SQLiteDatabase db = this.getReadableDatabase();
Cursor res = db.rawQuery("SELECT " + ENTRY_COLUMN_ID + ", " + ENTRY_COLUMN_TITLE + ", " + ENTRY_COLUMN_DATE + ", " + ENTRY_COLUMN_TIME + ", " + ENTRY_COLUMN_ICON + ", " + ENTRY_COLUMN_NOTES + " FROM " + ENTRY_TABLE_NAME + " ORDER BY date(" + ENTRY_COLUMN_DATE + ") DESC, time(" + ENTRY_COLUMN_TIME + " DESC", null);
```

Snippet 8. Querying the database

2) To customise the list, a XML layout file was created to specify a custom view for diary entry items of the list (see Figure 16). This view contains text and image containers to represent entries’ data such as title, date, time or icon showing how positive users were while experiencing the entry event. This custom item list layout was also used in the `HomeFragment` list.

3) The fragment also allows to navigate to another activity (`OneEntryActivity`) showing entry’s full details when clicking in one of the entries.

Figure 16. First prototype’s EntriesFragment screen
3.2.2. NewEntryActivity

When users click on the floating “+” button in MainActivity or over any of the points in the main chart, they are creating an Intent to display NewEntryActivity.

1) This activity generates a UI screen to create a new diary entry (see Figure 17). It provides date and time picker dialogs to specify when the event happened, two editable text inputs to indicate title and some notes, and a group of icons to express event’s valence. If the user clicks on one of the main chart’s points, date and time input fields would be automatically completed.

2) Except for the “notes” input field, all the fields are required to create a new entry. If the required fields are completed and user clicks on the confirmation button, a new row will be inserted in the corresponding database table. As other database operations, NewEntryActivity utilises a method belonged to MSBandDiaryDBHelper.

3) This activity is also used to edit existing diary entries. When a user selects “Edit” button on OneEntryActivity (explained below), all the input files are completed with the corresponding entry. A “tick” button clicking triggers an updating database operation.

![Figure 17. First prototype’s NewEntryActivity screen](image)

3.2.3. OneEntryActivity

When users click on an entry item in one of the lists, they are creating an Intent to display OneEntryActivity.
1) Basically, this activity displays all the information stored about a diary entry (see Figure 18). Its layout file contains different UI elements which are edited with a given entry’s information retrieved from the database.

2) OneEntryActivity also provides two different buttons. The first one deletes the entry being viewed after user confirmation. The second one allows the user to edit that entry using the NewEntryActivity.

3.2.4. SendFeedbackActivity

An Intent to display SendFeedbackActivity is created when clicking on the “Feedback” button found in the MainActivity (see Figure 19). This activity was created to provide a tool which allowed to get immediate feedback from subjects testing the application.

1) It displays a simple screen containing two editable text inputs and informs users about what they can use them for.

2) Once the “tick” button is clicked, it shows the existing e-mail client to select. Then, the recipient, subject and body fields will be filled automatically on the user’s e-mail client selected.
3.3. Broadcast receivers

MSBand Diary implements four different broadcast receivers since its first prototype. They are triggered as a response to multiple announcements generated by both MSBand Diary or Android system. More specifically, Broadcast Receivers take part of a process to set repetitive time-based operations or those after the Android booting process.

Scheduling time-based operations is a task for Android alarms (based on Android AlarmManager class). Outside the lifetime of the application, they invoke long-running operations even when the application is not running or in sleep mode, such as starting a service every 30 seconds or setting a reminder every day at the same time.

3.3.1. OnBootReceiver

Regardless of whether the application is opened by the user, this class implementing BroadcastReceiver class receives a broadcast after the system finishes booting. It requires a special permission specified in the manifest file called RECEIVE_BOOT_COMPLETED and an intent filter that filters on the ACTION_BOOT_COMPLETED (see Snippet 9).

```xml
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />

// ...

<receiver
    android:name=".OnBootReceiver"
    android:enabled="true"
    android:exported="false">
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED" />
    </intent-filter>
</receiver>
```

Snippet 9. Permission to receive broadcast after system booting
Once system finishes booting, this `OnBootReceiver` receives the announcement and sets three different alarms which schedule calls to the other three broadcast receivers in the application at given times (see Snippet 10). It is worth noting that `AlarmManager` can work with different types of alarm. In our case, `ELAPSED_REALTIME_WAKEUP` fires the alarm every time interval specified since current time and `RTC_WAKEUP` uses a concrete time.

```
package lg21.macs.hw.ac.uk.msbanddiary;
import ...;

public class OnBootReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent.getAction().equals("android.intent.action.BOOT_COMPLETED")) {
            Intent alarmIntent = new Intent(context, MSBandDataReceiver.class);
            PendingIntent pendingIntent = PendingIntent.getBroadcast(context, 0, alarmIntent, 0);
            AlarmManager manager = (AlarmManager) context.getSystemService(Context.ALARM_SERVICE);
            // 30 seconds in millis
            int interval = 30000;
            // It fires the pending intent which starts the service after the // specified interval
            manager.setInexactRepeating(AlarmManager.ELAPSED_REALTIME_WAKEUP, SystemClock.elapsedRealtime(), interval, pendingIntent);
            Intent reminderIntent = new Intent(context, ReminderReceiver.class);
            PendingIntent pIReminder = PendingIntent.getBroadcast(context, 0, reminderIntent, PendingIntent.FLAG_UPDATE_CURRENT);
            Calendar calendar = Calendar.getInstance();
            calendar.set(Calendar.HOUR_OF_DAY, 21);
            calendar.set(Calendar.MINUTE, 0);
            calendar.set(Calendar.SECOND, 0);
            calendar.set(Calendar.MILLISECOND, 0);
            if(calendar.getTime().compareTo(new Date()) < 0)
                calendar.add(Calendar.DAY_OF_MONTH, 1);
            // Fires the intent which set the notification reminder every 24 // hours starting at time specified in calendar (21:00)
            manager.setInexactRepeating(AlarmManager_RTC_WAKEUP, calendar.getTimeInMillis(), AlarmManager.INTERVAL_DAY, pIReminder);
            Intent resetDBIntent = new Intent(context, ResetDBReceiver.class);
            PendingIntent pIResetDB = PendingIntent.getBroadcast(context, 0, resetDBIntent, PendingIntent.FLAG_UPDATE_CURRENT);
            Calendar startResetDB = Calendar.getInstance();
            startResetDB.set(Calendar.HOUR_OF_DAY, 0);
            startResetDB.set(Calendar.MINUTE, 0);
            startResetDB.set(Calendar.SECOND, 0);
            startResetDB.set(Calendar.MILLISECOND, 0);
            if(startResetDB.getTime().compareTo(new Date()) < 0)
                startResetDB.add(Calendar.DAY_OF_MONTH, 1);
            // Fires the intent which resets the DB every 24 hours starting at // time specified in calendar (00:00)
            manager.setInexactRepeating(AlarmManager_RTC_WAKEUP, startResetDB.getTimeInMillis(), AlarmManager.INTERVAL_DAY, pIResetDB);
        }
    }
}
```

Snippet 10. `OnBootReceiver` class
A code similar to the one inside the `onReceive` method is included in the `MainActivity` class to start the service and set the notification and database reset when the user opens the application for the first time, since `OnBootReceiver` is only used after system booting.

### 3.3.2. MSBandDataReceiver

Basically, this broadcast receiver starts the application service when it receives the broadcast (see Snippet 11). It is invoked every 30 seconds.

```java
package lg21.macs.hw.ac.uk.msbanddiary;
import ...

public class MSBandDataReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        Intent serviceIntent = new Intent(context, MSBandDiaryService.class);
        context.startService(serviceIntent);
    }
}
```

Snippet 11. `MSBandDataReceiver` class

### 3.3.3. ResetDBReceiver

As explained previously, the application needs to reset all the stored values for user’s heart rate and Galvanic Skin Response data at the end of each day. Towards this end, an alarm firing `ResetDBReceiver` is activated every 24 hours at 00:00 (see Snippet 10).

`ResetDBReceiver` simply gets an instance of the `MSBandDiaryDBHelper` and executes one of its methods, which updates all the heart rate and GSR values to 0.

### 3.3.4. ReminderReceiver

In order to keep users active writing their diaries, an Android notification was sent every day at 21:00 in the application’s first prototype. `ReminderReceiver` is responsible for creating these notifications (see Figure 20). A click on the notification opens MSBand Diary’s home screen.
As explained later in this report, the default notification time (21:00) can be changed in the application's second prototype.

3.4. Services

MSBand Diary’s full functionality relies on MSBandDiaryService, a Service implementation running in the background of the application. It is started by means of MSBandDataReceiver every 30 seconds to update heart rate and GSR data in the database.

Consequently, MSBandDiaryService needs to subscribe to several Microsoft Band’s sensors in order to retrieve information from them. Apart from those receiving heart rate and GSR data, it also uses a “Contact” sensor which determines if the band is being worn or not.

After checking if Microsoft Band is connected and paired with the Android device, MSBandDiaryService registers a listener to know if the bands is being worn (see Snippet 12). If so, and the user has already accepted the consent to get data from heart rate sensor, it also register other two listeners to get heart rate and GSR. Inside these listeners, MSBandDiaryDBHelper methods are used to update the corresponding table.

If any error occurs during the service execution, the user receives an error when application is opened.
private BandContactEventListener contactEventListener = new BandContactEventListener() {

    @Override
    public void onBandContactChanged(final BandContactEvent event) {
        if(event!=null){
            BandContactState state = event.getContactState();
            if(state != BandContactState.NOT_WORN){
                if(client.getSensorManager().getCurrentHeartRateConsent() == UserConsent.GRANTED) {
                    try {
                        client.getSensorManager().registerHeartRateEventListener(heartRateListener);
                        client.getSensorManager().registerGsrEventListener(gsrEventListener);
                    } catch(BandException ex) {
                        Toast.makeText(getApplicationContext(), "Heart rate consent not accepted", Toast.LENGTH_SHORT).show();
                    } catch(InvalidBandVersionException e) {
                        e.printStackTrace();
                    }
                }
            }
        }
    }
};

Snippet 12. Band’s Contact sensor listener

3.5. Resources

In terms of resources, each activity or fragment needed an XML layout file where to declare all the UI components that will be used later to create interaction with users. Navigation drawer and menus’ functioning also depends on their XML layout (see Snippet 13).

```xml
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <group android:checkableBehavior="single">
        <item android:id="@+id/nav_home"
              android:icon="@drawable/ic_home"
              android:title="My Diary"/>
        <item android:id="@+id/nav_list"
              android:icon="@drawable/ic_view_list"
              android:title="Entries list"/>
    </group>
</menu>
```

Snippet 13. First prototype’s Navigation Drawer’s XML layout file

As can be seen in Snippet 13, the application also contains a number of drawable images that can be referenced from code. Some of them are provided by Android, but other were specially created for this application, such as the “positivity” icons (see Table ).
4. MSBand Diary’s second prototype

As an Evolutionary Prototyping project, the first prototype forms the core of the system, and further prototypes include improvements and new requirements. Most of these updates were the outcome of the usability evaluation discussed later in this report, where users gave their feedback about the first prototype and suggested new functionalities. The second prototype also included solutions to performance problems detected directly by the developer and other optimisation updates.

MSBand Diary’s second prototype required the inclusion of new activities and fragments, as well as a set of resources to create the user interface’s new features. Those aforementioned Android components which did not suffer modifications will not be explained again in this second prototype.

4.1. Database support

*MSBandDiaryDBHelper* included a new table to store emotions names, as application enabled user to manage a emotions list. By default, the list includes 6 different emotions according to the primary emotions described in Shaver et al. (1987): Love, Joy, Surprise, Anger, Sadness and Fear. Therefore, the database helper needs to initialise the table with those rows (see Snippet 14), as well as providing methods to create and delete emotions.
Furthermore, users can now include pictures and emotions in their entries, therefore entry table was modified adding two new columns: one to store the picture’s path in the device and other for several emotion names separated with commas.

4.2. Activities

This second prototype adds two news activities and three new fragments to the application. A summary of these activities can be found in Table 6.

<table>
<thead>
<tr>
<th>Activity name</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>MainActivity</td>
<td>Launcher activity which use fragments (in this prototype, 5 fragments) to show different screens.</td>
</tr>
<tr>
<td>NewEntryActivity</td>
<td>Activity which generates a UI screen to create a new diary entry providing several inputs fields. It is also used to edit past entries. In this prototype, it also allows to add images and emotions’ names.</td>
</tr>
<tr>
<td>OneEntryActivity</td>
<td>It displays all the information stored about a diary entry.</td>
</tr>
<tr>
<td>SendFeedbackActivity</td>
<td>It provides a screen which allowed to get immediate feedback from subjects by sending emails.</td>
</tr>
<tr>
<td>CreateEmotionsActivity</td>
<td>It displays a list of current emotions stored in the database, where users can remove them after confirmation, or create new ones entering their names.</td>
</tr>
<tr>
<td>SettingsActivity</td>
<td>This activity allows users to disable daily notifications or change the time they are triggered.</td>
</tr>
</tbody>
</table>

Table 6. Second prototype’s activities

Snippet 14. Emotions table inicialisation

```java
public void initialiseEmotionTable(SQLiteDatabase db){
    ContentValues contentValues;
    String[] emotions = new String[] {"Afraid", "Angry", "Sad", "Happy", "Surprised", "Loving"};
    for(String s : emotions){
        db.beginTransaction();
        try {
            contentValues = new ContentValues();
            contentValues.put(EMOTION_COLUMN_NAME, s);
            db.insert(EMOTION_TABLE_NAME, null, contentValues);
            db.setTransactionSuccessful();
        } catch (Exception e) {
            Log.d(TAG, "Error while trying to create Emotion table");
        } finally {
            db.endTransaction();
        }
    }
}
```
4.2.1. MainActivity

Expanding the application’s functionalities, MainActivity now adds three new options on the Navigation Drawer (see Figure 21). They allow users to display all their entry pictures using a grid view, and visualise weekly/monthly summaries of their entries based on emotions and positivity in entries. For this purpose, it uses different fragments for each choice.

EntriesFragment

At this second prototype, this fragment displays the entries in a given week found in the database in a custom list sorted by date. The set of entries is established directly using SQL code in the corresponding MSBandDiaryDBHelper class method, which receives two String objects specifying start and end date in YYYY-mm-dd format (see Snippet 15).

```java
public Cursor getCompleteEntriesBetween(String start, String end) {
    SQLiteDatabase db = this.getReadableDatabase();
    String[] params = new String[] { start, end };;
    Cursor res = db.rawQuery("SELECT " + ENTRY_COLUMN_ID + ", " + 
            ENTRY_COLUMN_TITLE + ", " + 
            ENTRY_COLUMN_DATE + ", " + 
            ENTRY_COLUMN_TIME + ", " + 
            ENTRY_COLUMN_ICON + ", " + 
            ENTRY_COLUMN_NOTES + " FROM " + ENTRY_TABLE_NAME + " WHERE " + 
            ENTRY_COLUMN_DATE + " BETWEEN ? AND ? ORDER BY date(" + ENTRY_COLUMN_DATE + ") DESC, time(" + ENTRY_COLUMN_TIME + ") DESC", params );
    return res;
}
```

Snippet 15. Retrieving entries data within a given time period

The week is provided by user using new UI components in the fragment (see Figure 22).
WeeklySummaryFragment and MonthlySummaryFragment

In early usability questionnaires, subjects were asked about adding extra screens to display summaries of its entries in a graphical way. This would help them to simply group their entries and get an overall view of their weeks or months in terms of emotions and positivity.

Two fragments were designed to enable users to input a week or month and display two different charts showing statistical data from entries (see Figure 23 for weekly example).

GalleryFragment

GalleryFragment filters all the entries containing a picture in the database and displays these pictures in a grid view sorted by date. To keep the UI's smoothness, the pictures are loaded in
a different thread to the one showing the UI by means of Universal Image Loader library (Sergey Tarasevich's Github page, 2011). A click on one of these images starts a OneEntryActivity showing all the data corresponding to its entry. The set of entries is established directly using SQL code in the corresponding MSBandDiaryDBHelper class method.

4.2.2. NewEntryActivity

Several of application’s new features were included in this activity:

1) As already mentioned, the second prototype allows users to include pictures in their entries. For that purpose, a new button is set in the UI to display a dialog where user can select if choosing an image from Android gallery or taking a photo using the camera. Once the picture is obtained, its complete path is saved to be further stored in the database.

2) The possibility of select “emotions” was also incorporated at this prototype. In this way, users can keep a diary and track their mood without needing to even write. Another button was added to show an editable list of emotions where users can choose those more appropriates.

3) Lastly, users’ selections about emotions and positivity in entries are now turned into text, making users more aware of what they are adding in the entry.
The XML layout file for NewEntryActivity was therefore modified to enable all these new operations while user is interacting with the system (see Figure 24).

4.2.3. CreateEmotionsActivity

When user wants to add emotions to an entry, a list is displayed for their selection. However, this list can be modified if desired and, for that purpose, there is an “Edit” button in the list. This button starts a CreateEmotionsActivity.

This activity displays a list of current emotions stored in the database. Basically, users can remove them after confirmation, or create new ones entering their names. The changes will be updated using MSBandDiaryDBHelper methods and can be immediately used when creating a new entry.

4.2.4. SettingsActivity

By default, the application’s first prototype generated a daily notification at 21:00. However, several subjects expressed their disagreement about either receiving notifications or the default notification time (see point 1.2 in the Evaluation section). For that reason, SettingsActivity was implemented.

This activity essentially allows users to disable daily notifications or change the time they are triggered. To store this user information, MSBand Diary uses Android Preferences, an Android mechanism to store pair of values for certain users’ preferences. In our case, when the user changes a notification setting, the system automatically updates the value in the SharedPreferences file. Therefore, where daily notification were activated (MainActivity and OnBootReceiver), it is now necessary to check these user’s preferences (see Snippet 16).

SettingActivity is started when user selects “Setting” in the application menu. This menu also includes another option called “About” which displays a dialog explaining the purpose of this project and how to interpret the main chart.
5. Testing and limitations

The beginning of this dissertation was delayed due to a developer’s medical problem, and therefore, time periods for each implementation were also reduced. This led to the development of two prototypes (instead of three as initially planned). Despite this, the software development process reached the milestones defined in advance for the first two prototypes, and those functionalities which were left undone will be included further in Future Work section.

During the complete process, the application was constantly tested by the developer, as well as by different subjects once the first prototype was completed. Testing helped the application to be debugged and the identification of several application’s limitations.

In some occasions, application responded differently depending on the Android version running on the mobile phone. Modification of certain parts of the code was necessary to face these problems, and Android 4.2 (API 17) was established as the minimum Android version supported by MSBand Diary.

The application also requires that the Microsoft Band and the Android device are permanently connected via Bluetooth, resulting in an extra battery consumption by Android phone. Moreover, both devices must stay in a range of 10 metres for proper application’s functioning. If not, the MainActivity’s chart may show null values heart rate or arousal level at some points.
The evaluation approach for this dissertation was divided in two different phases. The first of them was designed to evaluate the usability of the application’s first prototype, as well as the collection of potential requirements to be included in the second prototype. The second phase was focused on the application of qualitative evaluation methods in the second (and last) prototype, in order to know whether the application fulfils the main objectives described in previous sections. Furthermore, it contained a final usability survey for the application’s second prototype.

Due to the nature of this project, some legal and ethical issues were considered. The questionnaire data for evaluation and other data obtained from participants in the prototypes’ evaluation were anonymised by disguising identifiable data. Still, at the beginning of evaluation periods, a form was provided to the participants informing about the aim of this dissertation and what activity they are meant to perform during the evaluation of the prototypes. This form can be found in the Appendix D section of this paper. Furthermore, working with subjects’ emotional side occasionally leads to the system collecting and showing specific aspects of the user’s behaviour that may not be accepted or correctly interpreted by users, provoking rejection, as users may not be prepared to learn or be aware of certain behaviours in their own emotions. For that reason, users were informed that they could stop the experiment at any time they wish without giving any reasons.

Participants in both evaluation parts were contacted personally, as any user would need the application installed in an Android device connected with a Microsoft Band. These requirements do not allows us to contact people indirectly because of the low probability to find potential users possessing a Microsoft Band.

Both evaluation sections have a face-to-face interview, and a questionnaire which users were asked to complete later. Face-to-face situations helped to visually analyse prototypes’ usability, detecting possible issues on user interfaces which will need to be fixed in further versions.
1. Usability interviews (Evaluation Part 1)

The evaluation of this project mainly consists of carrying out a qualitative evaluation plan to determine whether the application supports users’ self-reflection on past events by observing own physical reactions. However, it was necessary to obtain users’ feedback from early stages which helped us to build a successful application in terms of user experience and performance, which motivates users to interact with the system.

The qualitative evaluation was performed by three users due to the number of Microsoft Bands available in this project. Nevertheless, the usability interview could be made by a larger number of subjects, since it does not require so much time. Once users feel comfortable using the first prototype during some minutes, they could give us useful feedback for further application versions.

1.1. Material and methods

The number of subjects interviewed was eight - four males and four females participants between 24 and 32 years of age selected from different backgrounds. All of them were regular Android users. Three of those participants also took part during the second part of the application’s evaluation.

First of all, participants were given a detailed description of the project, and were briefly guided through the application in order to make them familiar with its features. After that, they were asked to perform some basic actions:

a. Explore the main chart and locate highest and lowest point for heart rate and arousal level (the application was populated with fake data in order to create the main chart).

b. Create a new diary entry.

c. Create a new diary entry in the highest arousal level point.

d. Visualise both entries using the Navigation Drawer.

e. Edit one of the entries changed any value.

f. Delete one of the entries.
After performing these actions, a questionnaire was sent to the participants so that they could provide feedback regarding the usability of the application. The questionnaire can be found in the Appendix C section.

1.2. Results

The first set of questions were related to general satisfaction using MSBand Diary’s UI. The results obtained are showed in Table 7 on a 5-point scale being 1 a total disagreement with the statement.

<table>
<thead>
<tr>
<th>Statement</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>I found the application MSBand Diary useful.</td>
<td>0/8</td>
<td>0/8</td>
<td>3/8</td>
<td>3/8</td>
<td>2/8</td>
</tr>
<tr>
<td>It was easy to learn to use MSBand Diary</td>
<td>0/8</td>
<td>0/8</td>
<td>0/8</td>
<td>2/8</td>
<td>6/8</td>
</tr>
<tr>
<td>MSBand Diary was simple to use</td>
<td>0/8</td>
<td>0/8</td>
<td>1/8</td>
<td>2/8</td>
<td>5/8</td>
</tr>
<tr>
<td>The interface of MSBand Diary is pleasant.</td>
<td>0/8</td>
<td>0/8</td>
<td>0/8</td>
<td>1/8</td>
<td>7/8</td>
</tr>
<tr>
<td>The interface components are well-organised and help to complete the different scenarios.</td>
<td>0/8</td>
<td>0/8</td>
<td>2/8</td>
<td>4/8</td>
<td>2/8</td>
</tr>
<tr>
<td>The information provided for MSBand Diary is easy to understand and its organisation is clear</td>
<td>0/8</td>
<td>1/8</td>
<td>2/8</td>
<td>3/8</td>
<td>2/8</td>
</tr>
<tr>
<td>MSBand Diary gives error message telling how to fix a problem</td>
<td>0/8</td>
<td>2/8</td>
<td>2/8</td>
<td>3/8</td>
<td>1/8</td>
</tr>
</tbody>
</table>

Table 7. Usability survey - General questions

From the results, we can interpret participants agreed that user interface was clear, easy to learn and attractive in its first prototype. Most of them also found the application as a useful tool for its objective, although next prototype had to improve the system to show error messages. With an average of 4,1 out of 5 most of the users agreed with the above statements (see Chart 1 for question marks averages).
The second set of questions asked about specific features of the first prototype, such as Microsoft Band connection, the generation of the main chart and the creation of diary entries. The results are represented in Table 8, and question marks averages can be found in Chart 2.

<table>
<thead>
<tr>
<th>Statement</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>I found easy to connect the MSBand to the application</td>
<td>0/8</td>
<td>0/8</td>
<td>1/8</td>
<td>3/8</td>
<td>4/8</td>
</tr>
<tr>
<td>I like how the main chart shows the data</td>
<td>0/8</td>
<td>0/8</td>
<td>2/8</td>
<td>4/8</td>
<td>2/8</td>
</tr>
<tr>
<td>I understand the information provided by the main chart</td>
<td>0/8</td>
<td>1/8</td>
<td>4/8</td>
<td>2/8</td>
<td>1/8</td>
</tr>
<tr>
<td>The interaction with the chart is satisfying</td>
<td>0/8</td>
<td>0/8</td>
<td>3/8</td>
<td>3/8</td>
<td>2/8</td>
</tr>
<tr>
<td>I feel comfortable adding new entries to the diary</td>
<td>0/8</td>
<td>0/8</td>
<td>0/8</td>
<td>1/8</td>
<td>7/8</td>
</tr>
<tr>
<td>I can properly find previous entries in the diary</td>
<td>0/8</td>
<td>0/8</td>
<td>1/8</td>
<td>2/8</td>
<td>5/8</td>
</tr>
</tbody>
</table>

Table 8. Usability survey - Specific questions
Microsoft Band’s connection and operations related to diary entries were highly marked by participants. However, even not being poorly marked, the main chart’s features did not generate as good results as expected. For that reason, the second prototype included an “About/FAQs” button in its menu where it is explained how the main chart is built and the physiological data involved in its creation. Zoom on the chart was also enabled on the second prototype to improve user interaction. An average of 4.2 out of 5 means an overall satisfaction on the application’s specific features.

The questionnaire also included two open-ended questions to state any other positive or negative aspect. The most relevant answers were as follows:

- The chart size was too small, which will make it difficult the interaction.
- Lack of a presentation screen and a “Help” section.
- Chart does not show data at certain periods.
- Beautiful, modern interface.
- Application provides a easy system to keep a diary on-the-go

The inclusion of “About/FAQs” section and the possibility of zooming in on the chart already solved some of the negative aspects presented in these open-ended questions. Errors showing data in the chart resulted from the frequent Microsoft Band’s exchange between participants’ wrists, which led to blank periods of time without retrieving any physiological values.

Last part of the usability survey asked users to give feedback about potential future features, using both answering given questions and suggesting for improvements. The results are showed in Table 9, including questions’ marks average in Chart 3.
Among the different suggestions for improvements given in the open-ended question, we can highlight the following:

- Horizontal screen orientation support
- Possibility of adding pictures to entries
- Ability to remove daily notification
- Microsoft Band receiving notifications about high levels of arousal or heart rate.
- A date picker to display diary entries in a given period of time.
- The creation of backups in the cloud

<table>
<thead>
<tr>
<th>Statement</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>I would find useful to add media content, such as pictures and video, in the entries creation.</td>
<td>1/8</td>
<td>1/8</td>
<td>4/8</td>
<td>1/8</td>
<td>1/8</td>
</tr>
<tr>
<td>I would like to include a location in diary entries.</td>
<td>2/8</td>
<td>2/8</td>
<td>2/8</td>
<td>1/8</td>
<td>1/8</td>
</tr>
<tr>
<td>I would like to receive notifications more frequently, for example, suggesting to write entries.</td>
<td>2/8</td>
<td>0/8</td>
<td>3/8</td>
<td>2/8</td>
<td>1/8</td>
</tr>
<tr>
<td>I would find useful displaying weekly/monthly summaries of entries based on feelings or positivity.</td>
<td>0/8</td>
<td>1/8</td>
<td>3/8</td>
<td>2/8</td>
<td>2/8</td>
</tr>
</tbody>
</table>

Table 9. Usability survey - Further versions questions

Chart 3. Further versions’ questions - Marking average

Among the different suggestions for improvements given in the open-ended question, we can highlight the following:

- Horizontal screen orientation support
- Possibility of adding pictures to entries
- Ability to remove daily notification
- Microsoft Band receiving notifications about high levels of arousal or heart rate.
- A date picker to display diary entries in a given period of time.
- The creation of backups in the cloud
These results led to consider several improvements for the next prototype, as well as the exclusion of several of the initial requirements defined, such as the use of GPS to locate where an experience happened. Decisions made according to these results were as follows:

- The second prototype had to allow users to include pictures in diary entries. Video or other type of media content would be considered in further versions.
- Users had to be able to disable and configure the daily notification. Another kind of notifications would be implemented in future prototypes, including their integration in Microsoft Band,
- Use of GPS was discarded for second prototype, although it would still be considered in the future.
- Periodic summaries had to be implemented in the new application’s version.

Other results not considered in these decisions were stored to be evaluated in application’s future updates.

2. Project evaluation (Evaluation Part 2)

The second part of the evaluation was based on user participation, and it consisted of the use of different techniques to collect qualitative data. Evaluation methods described in Literature Review section highly motivated qualitative techniques selected for users’ evaluation.

Consequently, given that only one Microsoft Band was available during the development of this project, participants evaluated the system one at a time. For that reason, the application’s first prototype was released as soon as possible to obtain enough time to carry out the initial evaluation.

2.1. Material and methods

To be part of their everyday lives, participants were asked for testing the second application’s prototype for a period of 7-10 days. The number of participants that could take part in the study was limited due to the equipment that was lent to evaluate the system; an only Microsoft Band was available. Moreover, because of a medical problem suffered by the developer, all the time periods stipulated had to be reduced. This fact led to the purchase of
an extra Microsoft Band to carry out the final period of the project’s evaluation. Finally, the number of participants was three - one male and two female subjects between 27 and 32 years of age.

Before the beginning of the evaluation period (7-10 days), participants were instructed to learn how to use the new application’s prototype together with the Microsoft Band. At the end of the evaluation period, using a semi-structured interview (see Appendix C section), participants were interviewed to discuss about how positively they could match the data represented by the application with real experiences and whether to use the prototype was beneficial to record and reflect about particular events in their everyday lives.

This evaluation part also carried out another questionnaire, which could complement the interview, since users may become uneasy to say, for example, negative aspects of the systems in face-to-face conversations. Therefore, they included open-ended questions to invite users to freely express their opinions, as well as others related to system functionalities or user experience in the form of ranked questions. Moreover, a free general comment box was included to know any users’ suggestions related or not with the prototype.

The resulting data consisted of recordings of the interviews - and transcriptions of some of their parts; screenshots of any application state remarkable for the participant; and a set of final questionnaires.

2.2. Results

As discussed previously, the main objective of the evaluation was to determine whether the application supports users’ self-reflection on past events by observing own physical reactions. Additionally, we wanted to know how participants dealt with wearable sensors and a mobile-based diary, as well as the use they made of the different application’s resources.

2.2.1. Getting used to wearable sensors and a mobile-based diary

According to the use of wearable sensors, in our case the Microsoft Band, none of the participants was used to wear them, not even any kind of smart watch. As the correct
functioning of the system relies on the connection with sensors, we also wanted to know potential problems in their use.

Participants A and B, from backgrounds in Social Work and IT respectively, agreed that Microsoft Band was actually easy to wear, and participant B also showed great enthusiasm in other features provided by the Microsoft Band, such as the step counter. On the contrary, participant C, from a background in Psychology, was reluctant to wear it throughout the whole day during the study:

“Microsoft Band is uncomfortable to wear while working. It is not very flexible and too heavy. Moreover, it is not water-resistant.”

As for keeping the diary, all the participants filled in the diary actively during the study—considering that the average of entries created was 5 a day. Participant A stated that never having kept a diary, it was really easy to complete, mainly due to it was accessible from the mobile phone. This fact was also highlighted by the other two participants:

“I always have my phone with me, wherever I go. Writing down something important was really easy.”

“Pen and paper diaries need to be normally completed at some point in the day, before going to bed in my case. However, taking notes on the move makes you take advantage of any spare moment to write about recent experiences.”

There was no issue found in the use of virtual keyboard, as participants were familiar with it. Curiously enough, participant B that there was only a detail missed: the smell of the paper diary.

### 2.2.2. Input modes

Using different Android devices’ tools, our objective was to allow users to record different types of materials in their entries, which could be used to remember and reflect on past experiences. This had a positive effect, since participants often referred to the different inputs types during the interviews explaining the frequent use they made of them.

Media types offered by MSBand Diary enabled us to investigate how input modes could be used to capture particular piece of information. We found text, icons and emotions selected from the list as the most used input modalities by the three participants to capture thoughts
and experiences, although participant A considered pictures as a powerful tool to catch a moment’s context:

“Pictures are able to catch a context for certain experiences. Combining both text and pictures helped me to remember things while re-reading the diary”

2.2.3. Interpreting physiological data

Apart from recognition, it was key to investigate how participants reflected in and interpreted the chart in MSBand Diary. This reflection and interpretation varied between the different subjects. Sometimes participants related to the points in the chart, seeing the heart rate and arousal level as representative of their own past events.

First of all, participants made use of the diary for different purposes. Participant A collected experiences that were supposed to match with a high level of arousal, and then using the main chart, it could be checked if they matched or not. Appreciating physiological changes was the main motivation to create entries for participant A. Figure 25 shows a screenshot during participant A’s evaluation period. A was surprised due to the level of accuracy showed by the arousal level representation in the chart during a stressful work shift from 4-8pm.

![Figure 25. Participant A’s application screenshot](image)
On the other hand, both participants B and C noted all the experiences they considered, using the chart as a reminder tool. Participant C identified the process why entries were created in the diary:

“I discover moments of important changes in the chart when I thought that nothing had happened. Then, it made me write about experiences that I had ignored.”

However, the ability of representing past experiences on physiological data was not always supported. The data in chart were defined occasionally as inexact recreation of users’ arousal state. For instance, during the evaluation period, participant B strongly argued with a colleague in a job meeting. It was expected to produce relevant changes in the chart, but no strong evidence was found in MSBand Diary.

2.2.4. Self-reflection

Aside from interpreting and using several input modalities in the diary or re-living experiences, the participants experienced reflections on past events and other processes more reflective. Sometimes, the participants created direct relationships between the diary and their lifestyle. Other times, data representation in MSBand Diary was considered as simple data showed virtually, without understanding them as a part of their own lives.

Analysing the application’s results during the evaluation period, Participant C appeared to reflect on and learn from the diary’s content, probably due to participant’s background. C connects data provided by MSBand Diary with stress levels and emotions:

“The application allowed me to control my emotional state during the week while being aware of and identifying my sources of stress”

Despite the fact that participants were informed at the beginning of their evaluation period about arousal level does not always involve “bad” emotions, participant A also interpreted those levels as a stress symptom. A also highlights how application improves self-awareness about everyday activities:

“I could learn from the information [provided by MSBand Diary], for example what activities affect my body in a better or worse way, or the difference between a day at work and a day off.”

Participant B did not show potential signs of self-reflection during the interview, focusing most of the answers on the simple visualisation of data and emotions.
2.2.5. Conclusion on face-to-face interviews

A basic part of the evaluation was whether our participants understood the graphical representations for heart rate and arousal level. Overall, participants demonstrated they were able to do so. The ways in which the participants used the mobile phone materials with the chart data illustrated that the system provided successful tools to build their own mobile-based diaries.

We also found that participants frequently related to and interpreted their diaries in different ways. Arguably though, participant B’s example of not accurate interpretation revealed that MSBand Diary can still improve its system of data collection and representation.

At different levels, MSBand Diary enabled reflection on past events through the visual representations of participants’ recordings and graphical support. A clear example was the answer of one of participant C in the final evaluation interview:

“The application was really useful to make me aware of how feelings and emotions correlate with my body condition, enabling to control the way I face different situations every day”

However, as not all the participants experimented reflective processes, we consider that the application could still incorporate new features to motivate reflections and other self-awareness processes.

2.2.6. Final usability survey

Last part of the evaluation asked users to give feedback about the second prototype and system performance, as well as potential future features and error to be taken into account in further work, using both answering given questions and suggesting for improvements. The result are showed in Table 10, as well as questions’ marks average in Chart 4.

<table>
<thead>
<tr>
<th>Statement</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>The final interface of MSBand Diary is intuitive and pleasant.</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>1/3</td>
<td>2/3</td>
</tr>
<tr>
<td>I can easily navigate through different diary entries</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>3/3</td>
</tr>
</tbody>
</table>
The inclusion of pictures in diary entries is made correctly
Pictures taken from the application are displayed properly when required
Weekly/monthly charts are created accurately to my entries
I found the application performance in my Android device satisfactory
I have not noticed any change in the battery performance of my device since I have been using MSBand Diary
MSBand Diary does not need a disproportionate amount of memory
Pictures taken from the application are stored properly in the external memory of my Android device
Overall, MSBand does not contain relevant bugs

Table 10. Usability survey - Further versions questions

<table>
<thead>
<tr>
<th>Statement</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>The inclusion of pictures in diary entries is made correctly</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>2/3</td>
<td>1/3</td>
</tr>
<tr>
<td>Pictures taken from the application are displayed properly when required</td>
<td>0/3</td>
<td>0/3</td>
<td>1/3</td>
<td>1/3</td>
<td>1/3</td>
</tr>
<tr>
<td>Weekly/monthly charts are created accurately to my entries</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>3/3</td>
</tr>
<tr>
<td>I found the application performance in my Android device satisfactory</td>
<td>0/3</td>
<td>0/3</td>
<td>1/3</td>
<td>0/3</td>
<td>2/3</td>
</tr>
<tr>
<td>I have not noticed any change in the battery performance of my device since I have been using MSBand Diary</td>
<td>0/3</td>
<td>0/3</td>
<td>1/3</td>
<td>1/3</td>
<td>1/3</td>
</tr>
<tr>
<td>MSBand Diary does not need a disproportionate amount of memory</td>
<td>0/3</td>
<td>1/3</td>
<td>1/3</td>
<td>1/3</td>
<td>0/3</td>
</tr>
<tr>
<td>Pictures taken from the application are stored properly in the external memory of my Android device</td>
<td>0/3</td>
<td>0/3</td>
<td>1/3</td>
<td>2/3</td>
<td>0/3</td>
</tr>
<tr>
<td>Overall, MSBand does not contain relevant bugs</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>1/3</td>
<td>2/3</td>
</tr>
</tbody>
</table>

Chart 4. Final usability questions - Marking average
Among the different suggestions for improvements given in the open-ended questions, we can highlight the following:

- Possibility of adding voice messages to entries
- Adding notifications when application detects a high level of arousal or heart rate
- Adding locations in entries
- Different challenges could be added to the application, in order to achieve, for example, certain number of positive entries

As for the errors or problems found in the system’s functionality, participants only mentioned those related to the aspect of wearing constantly the Microsoft Band having the Android device near, since otherwise chart shows no data.

Due to the number of participants taking part in this last usability survey (only three), results can only be indicative. However, the purpose of this last questionnaire was to get a light feedback about the last prototype, where we could identify serious errors and get future functionalities to be added in future work from a user’s point of view. If time had permitted, some of them had been included in a supposed third prototype.
Future work

In this MSc project, an Android mobile-based application was developed to improve user’s self-awareness and produce reflection on past events while interacting with the system. A set of tools were included to help users to identify and represent experiences, but due to time requirements, other tools were left out.

Regarding to input modality, the application could accept audio and video as a tool to catch experiences’ context or simply enable the possibility of create diary entries aloud. The use of Android GPS sensor could also be useful to identify experience’s location and therefore classify where relevant levels of arousal or heart rate occurred.

Project was implemented using Android development. However, a future cross-platform system could be developed, allowing users to complete their diaries from different devices (mobile phones, tablets, laptops…). Such a work would need to support remote databases instead of local database in current application, considering the privacy and security concerns that would come with it.

Machine learning techniques could be used to allow MSBand Diary to learn and find common patterns in user’s physiological data, also considering event’s time and place. For instance, providing the application with some frequent locations, it can suggest or interpret events such as a stressed situation at work or a challenging workout at gym.

Lastly, taking into account the positive feedback received related to the creation of diary entries, a potential future work could be the adaptation of MSBand Diary to Cognitive Behavioural Therapies (CBT). According to Pearson (2001), CBT studies treat a variety of mental illnesses, such bipolar disorders, anxiety, stress and depression. Its main aim is to change these behaviours by identifying negative and distorted thoughts, stating a link between thoughts, feelings and behaviour. Markway (2014) declares that writing down own thoughts is the best way to learn from them, and there is where diaries come in. In this case, they are called thought diaries, a documentation tool for monitoring feelings of, for example, anxiety, fear, sadness, anger or shame, also encouraging to use alternative thoughts. This self-
awareness process could be a suitable use of MSBand Diary since, in addition to provide tools to monitor thoughts and feelings, its graphical support could be beneficial to identify past thoughts and situations.
Conclusion

This project has been presented into three parts: a theoretical background in Affective Interaction, correlation between emotions and physiological correlates and mobile-based studies; a user-driven development of MSBand Diary system; and a qualitative study of the system using different evaluation methods. Through these sections, our overall aim was to explore the potential of a system that motivate users to recognise and reflect on past events by visualising their own physiological data.

To a certain extent, the developed application allowed us to achieve our objective. During the evaluation process, participants showed some signs of reflection and self-awareness using certain tools provided by MSBand Diary. Although one of the participants struggled to identify the data represented as own, the others seemed to be comfortable observing their physiological data to notice emotional states dairy and, in that way, learn from their own experiences.

However, the development of this study does not conclude at this point. Additional functionalities and tools will be included in further application versions, which will require more complete and extensive evaluation methods to explore deeper aspects on the impact of the system on users, until turning it into a complete, publicly available software. The evaluation plan carried out during this dissertation was considered successful, but longer evaluation periods will provide more detailed and sensible information about future work and potential application’s uses.
References

American Heart Association website, (2016). Target Heart Rates [online]. Available at: http://www.heart.org/HEARTORG/HealthyLiving/PhysicalActivity/FitnessBasics/Target-Heart-Rates_UCM_434341_Article.jsp#.V7QShjMrLPA [Accessed 22 July 2016]


Palen, L. and Salzman, M., 2002, November. Voice-mail diary studies for naturalistic data capture under mobile conditions. In Proceedings of the 2002 ACM conference on Computer supported cooperative work (pp. 87-95). ACM.


List of Figures

Figure 1. Affective diary: adding a note (extracted from Lindström et al., 2006) 13
Figure 2. Affective diary: adding a picture (extracted from Lindström et al., 2006) 13
Figure 3. Example of data collected in serendipity mobile-diary application (extracted from Sun et al., 2013) 15
Figure 4. Moodtrack Diary (Google Play Store) 15
Figure 5. In flow - Mood Diary (Google Play Store) 15
Figure 6. Evolutionary prototyping 20
Figure 7. System’s functional architecture 24
Figure 8. First prototype’s user interface 26
Figure 9. Second prototype’s user interface 29
Figure 10. Providing horizontal orientation while creating entries 30
Figure 11. Entity - Relationship database’s initial model 31
Figure 12. Last prototype’s Entity - Relationship model 32
Figure 13. MSBand Diary launcher icon 37
Figure 14. First prototype’s MainActivity screen 40
Figure 15. First prototype’s HomeFragment screen 42
Figure 16. First prototype’s EntriesFragment screen 43
Figure 17. First prototype’s NewEntryActivity screen 44
Figure 18. First prototype’s OneEntryActivity screen 45
Figure 19. Clicking “Send feedback” button 46
Figure 20. Clicking on notification 49
Figure 21. Second prototype’s MainActivity screen 53
Figure 22. Second prototype’s EntriesFragment screen 54
Figure 23. WeeklySummaryFragment’s screen 54
Figure 24. Second prototype’s NewEntryActivity screen 55
Figure 25. Participant A’s application screenshot 67
List of Tables

Table 1. Initial requirements 21
Table 2. Optional requirements 22
Table 3. Requirements obtained during evaluation period 22
Table 4. First prototype’s activities 40
Table 5. “Positivity” icons 51
Table 6. Second prototype’s activities 52
Table 7. Usability survey - General questions 60
Table 8. Usability survey - Specific questions 61
Table 9. Usability survey - Further versions questions 63
Table 10. Usability survey - Further versions questions 70
List of Snippets

Snippet 1. Manifest file 36
Snippet 2. Database Helper class 38
Snippet 3. MSBand table initialisation 39
Snippet 4. Getting last 5 diary entries 39
Snippet 5. UI component declaration 41
Snippet 6. Checking Microsoft Band connection 41
Snippet 7. Getting database data 42
Snippet 8. Querying the database 43
Snippet 9. Permission to receive broadcast after system booting 46
Snippet 10. OnBootReceiver class 47
Snippet 11. MSBandDataReceiver class 48
Snippet 12. Band’s Contact sensor listener 50
Snippet 13. First prototype’s Navigation Drawer’s XML layout file 50
Snippet 14. Emotions table initialisation 52
Snippet 15. Retrieving entries data within a given time period 53
Snippet 16. Setting daily notification after checking user’s preferences 57
# List of Charts

<table>
<thead>
<tr>
<th>Chart Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chart 1. Specific questions - Marking average</td>
<td>61</td>
</tr>
<tr>
<td>Chart 2. Specific questions - Marking average</td>
<td>62</td>
</tr>
<tr>
<td>Chart 3. Further versions’ questions - Marking average</td>
<td>63</td>
</tr>
<tr>
<td>Chart 4. Final usability questions - Marking average</td>
<td>70</td>
</tr>
</tbody>
</table>
Appendix A: UML Class Diagrams

For representative purposes, the class diagrams of the project are shown across several diagrams, because the resulting figure would most likely be illegible in case of showing them in only one.

Application overview: project class diagram

CreateEmotionsActivity: creation of emotions inside an editable list

<table>
<thead>
<tr>
<th>CreateEmotionsActivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>-ListView: ListView</td>
</tr>
<tr>
<td>-dBHelper: MSBandDiaryDBHelper</td>
</tr>
<tr>
<td>-emotions: ArrayList&lt;String&gt;</td>
</tr>
<tr>
<td>+CreateEmotionsActivity()</td>
</tr>
<tr>
<td>#onCreate(Bundle): void</td>
</tr>
<tr>
<td>+onOptionsItemSelected(MenuItem): boolean</td>
</tr>
</tbody>
</table>
EntriesFragment

EntriesFragment
- view: View
- dbHelper: MSBandDiaryDBHelper
- wrapper: LinearLayout
- pickWeek: EditText
- pickYear: EditText
- week: int
- year: int
- startDate: String
- endDate: String
- cursor: Cursor

+ EntriesFragment()
+ onCreate(Bundle): void
+ onCreateView(LayoutInflater, ViewGroup, Bundle): View
- getWeekDetails(int): String
- onActivityResult(int, Intent): void

ItemListOnClickListener
~ indexDB: int
+ itemListOnClickListener(int)
+ onClick(View): void

GalleryFragment

GalleryFragment
- view: View
- dbHelper: MSBandDiaryDBHelper
- wrapper: GridView

+ GalleryFragment()
+ onCreate(Bundle): void
+ onCreateView(LayoutInflater, ViewGroup, ViewGroup, Bundle): View
+ onActivityResult(int, int, Intent, Intent): void

ImageAdapter
- IMAGE_URLS: String[]
- inflate: LayoutInflater
- options: DisplayImageOptions

+ getLayoutParams(): int
+ getGroup(): int
+ getLabel(): int
+ getItemCount(): int
+ getView(int, View, ViewGroup): View

PictureHolder
~ imageView: ImageView
~ progressBar: ProgressBar
~ PictureViewHolder

MainActivity

MainActivity
- currentFragmentTag: String
- toolbar: Toolbar

+ MainActivity()
+ onCreate(Bundle): void
+ onActivityResult(int, int, Intent): void
+ onBackPressed(): void
+ onCreateOptionsMenu(Menu): boolean
+ onOptionItemSelected(Menu): boolean
+ onNavigationItemSelected(MenuItem): boolean
+ isMyServiceRunning(Context): boolean

MSBandDiaryApplication: simple class to define several app’s configurations

MSBandDiaryApplication
+ MSBandDiaryApplication()
+ onCreate(): void
+ initImageLoader(Context): void

Application’s Broadcast Receivers

MSBandDataReceiver
+ MSBandDataReceiver()
+ onReceiver(Context, Intent): void

OnBootReceiver
+ OnBootReceiver()
+ onReceive(Context, Intent): void

ResetDBReceiver
- dbHelper: MSBandDiaryDBHelper
+ ResetDBReceiver()
+ onReceive(Context, Intent): void

ReminderReceiver
+ ReminderReceiver()
+ onReceive(Context, Intent): void
### WeeklySummaryFragment

- `view`: View
- `dbHelper`: MSBandDiaryDBHelper
- `pickWeek`: EditText
- `pickYear`: EditText
- `count`: TextView
- `week`: int
- `year`: int
- `startDate`: String
- `endDate`: String
- `cursor`: Cursor
- `positivity`: int
- `positivityTypes`: String[]
- `positivityColors`: String[]
- `emotions`: HashMap<String,Integer>
- `chart`: PieChartView
- `data`: PieChartData
- `columnChart`: ColumnChartView
- `columnData`: ColumnChartData
- `hasLabels`: boolean
- `hasLabelsOutside`: boolean
- `hasCenterCircle`: boolean
- `hasLabelForSelected`: boolean
- `hasAxes`: boolean
- `hasAxesNames`: boolean
- `dataType`: int

+ `WeeklySummaryFragment()`
+ `onCreate(Bundle); void`
+ `onCreateView(LayoutInflater,ViewGroup,Bundle); View`
+ `getStatistics(); void`
+ `generateCharts(); void`
+ `getWeekDetails(int,int);String`
+ `onActivityResult(int,int,Intent); void`

### MonthlySummaryFragment

- `view`: View
- `dbHelper`: MSBandDiaryDBHelper
- `pickMonth`: EditText
- `pickYear`: EditText
- `count`: TextView
- `month`: int
- `year`: int
- `startDate`: String
- `endDate`: String
- `cursor`: Cursor
- `positivity`: int
- `positivityTypes`: String[]
- `positivityColors`: String[]
- `emotions`: HashMap<String,Integer>
- `chart`: PieChartView
- `data`: PieChartData
- `columnChart`: ColumnChartView
- `columnData`: ColumnChartData
- `hasLabels`: boolean
- `hasLabelsOutside`: boolean
- `hasCenterCircle`: boolean
- `hasLabelForSelected`: boolean
- `hasAxes`: boolean
- `hasAxesNames`: boolean

+ `MonthlySummaryFragment()`
+ `onCreate(Bundle); void`
+ `onCreateView(LayoutInflater,ViewGroup,Bundle); View`
+ `getStatistics(); void`
+ `generateCharts(); void`
+ `getMonthDetails(int,int);String`
+ `onActivityResult(int,int,Intent); void`
Appendix B: Initial storyboards

As for user interface, two complete storyboard were designed to obtain an idea about the user interface’s appearance before the beginning of the application’s development. A storyboard is a technique to show interaction between users and systems, which includes a series of sketches or pictures giving examples of how the story takes place.

First prototype’s storyboard
Second prototype’s storyboard

Presentation interface

Login interface

Initial points

First use

“Create-user” interface

“Create-profile” interface

Main interface

“Select-address” interface
Appendix C: Questionnaires and interview for project evaluation

The different tools utilised during the application’s evaluation can be found in this appendix. They are divided in two parts: the first one was used for the first prototype’s usability evaluation, and the second one for final project’s evaluation (based on the application’s second prototype).

Evaluation part 1: Usability questionnaire

MSBand Diary Application
Usability survey

Please, mark the interval questions on a 5-point scale being 1 a total disagreement with the statement.

<table>
<thead>
<tr>
<th>General questions</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. I found the application MSBand Diary useful.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. It was easy to learn to use MSBand Diary</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. MSBand Diary was simple to use</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. The interface of MSBand Diary is pleasant.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. The interface components (buttons, navigation drawer…) are well-organised and</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>help to complete the different scenarios.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. The information provided for MSBand Diary is easy to understand and its</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>organisation is clear.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. MSBand Diary gives error messages telling how the fix the problem.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Specific questions</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>8. I found easy to connect the MSBand to the application.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9. I like how the main chart shows the data.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10. I understand the information provided by the main chart.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11. The interaction with the chart is satisfying.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12. I feel comfortable adding new entries to the diary.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13. I can properly find previous entries in the diary.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Open-ended questions

14. Please, state any negative aspect (problems, errors...) you find in the application’s functionality.

15. Please, state any other positive aspect you find in the application.

Further versions

16. I would find useful to add media content, such as pictures or video, in the entries creation.

17. I would like to include a location in diary entries.

18. I would like to receive notification more frequently, for example, suggesting to write entries.

19. I would find useful displaying weekly/monthly summaries of entries based on feelings or positivity.

20. Please, make suggestions for improvements in further application versions.
Evaluation part 2: semi-structured face-to-face interview

MSBand Diary Application
Project evaluation - Structured interview

The interviews are conducted face to face. Before starting, subjects are informed about the purpose of the interview and asked for the approval of being recorded.

1. Have you got experience using Android devices?
   a. If so, how often do you use it during a day?
2. And wearable sensor such as Microsoft Band or any other smart watch?
   a. If not, was it difficult to wear it constantly?
3. Before using MSBand Diary, had you ever kept a diary?
   a. If so:
      i. How easy was to keep a diary in a mobile phone?
      ii. What advantages or disadvantages do you consider it has compared to pen and paper diaries?
      iii. Is the device keyboard easier or harder for you than hand writing?
   b. If not:
      i. How difficult was to keep a diary?
      ii. How often did you use it?
      iii. What advantages or disadvantages could it present against pen and paper diaries?
4. What have you used the diary for? What kind of data have you captured?
5. What media type did you use more often to catch information in your diary? Icons, text, pictures, combinations...
6. What made you create new entries?
7. How useful was the chart showing your data correlates during the whole day?
8. Do you think it was accurate?
9. How positive or negative was it to know that information?
10. How could the application contribute to your everyday life?
11. Would you continue using MSBand Diary?
12. In your opinion, what was the best thing of MSBand Diary?
13. And the worst one?
# Project evaluation – Final usability survey

*Please, mark the interval questions on a 5-point scale being 1 a total disagreement with the statement.*

<table>
<thead>
<tr>
<th>Application's last prototype</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The final interface of MSBand Diary is intuitive and pleasant.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. I can easily navigate through different diary entries.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. The inclusion of pictures in diary entries is made correctly.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Pictures taken from the application are displayed properly when required.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Weekly/monthly charts are created accurately to my entries.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System performance</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>6. I found the application performance in my Android device satisfactory.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. I have not noticed any change in the battery performance of my device since I have been using MSBand Diary.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. MSBand Diary does not need a disproportionate amount of memory.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9. Pictures taken from the application are stored properly in the external memory of my Android device.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10. Overall, MSBand is bug-free.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Open-ended question</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>11. Did you spot any problems or errors in the system's functionality? If yes, please write them down</td>
<td></td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>12. Please, make suggestions for improvements in future application versions.</td>
<td></td>
</tr>
</tbody>
</table>
Appendix D: Ethical approval and human consent

As stated in ethical principles for research involving human participants, every participant taking part in a project evaluation should be informed about the nature of the research and give written consent prior any interview. This appendix includes the Human Consent form used in this research and its Ethical Approval form.

Human Consent form

Human Subject Consent

Name:

Project: MSBand Diary Application

I have been asked to participate as a subject in this project. The nature of the tasks I will be asked to perform have been explained to me in writing.

I understand that I can withdraw my participation at any stage without prejudice.

I understand that all data collected will be anonymised before any publication, unless my prior consent is sought.

Signed:

Date:

Dear participant

I would like to express my gratitude in your decision to take part to my research by accepting to use the application that you are going to evaluate and completing further interviews and questionnaires.

In this interview with the developer you will be informed about how to use the application and asked to perform some actions on the application "MSBand Diary" in order to test its usability. After a week, a questionnaire that the developer will present to you should be taken. A final interview has to be completed after you use of the application. This is done for the evaluation phase of our research and the development of our project. The time required for these actions will not exceed 15 minutes of your time.

Both of these questionnaires will be completed by you, if you agree, as an anonymous user and the data collected from the app will be anonymised. You should complete the questionnaires being fair and impartial.

Thank you again for your help
Ethical Approval form

School of Mathematical and Computer Sciences
Protocol for Ethics Approval

1. Title of research:
   Microsoft Band Monitor

2. Purpose of study:
   Testing and evaluating mobile applications’ prototypes

3. Is ethical approval required by another body linked to the research?
   No.
   
   If YES please attach copies of the approval given to the other body, and confirmation that no changes
   have been made to the protocol since approval was granted

4. Is permission required from another body to use data or research materials?
   No.
   
   If YES please attach copies

5. Does the research involve the use of human subjects:
   Yes. Subjects will be asked to test application’s prototypes, providing feedback by
   questionnaires. Some observational methods will be also carried out.
   
   If YES what is the nature of the research e.g. focus group, questionnaire, etc
   
   If NO please go to Q.13
   
   N.B. The researcher should have considered the use of secondary data sources and
   should be clear that the aims of research cannot be met without new primary research
   involving people.

6. Is written consent to be obtained?
   Yes. The written consent will be obtained at the beginning of evaluation period. It
   basically consists of information about the evaluation plan the subject is involved
   and the commitment the subject makes during the plan.
   
   If YES please attach a copy of the consent and information form or indicate when it will be supplied
   
   If NO please justify.
7. How long will a subject have to decide whether to take part in the study. 

7 days.

Time in days [ ]

If less than 1 day, please comment.
(Note that it is common in the case of face to face interviews not to give significant advanced notice. This is acceptable in view of maximising the response rates and reliability of some survey based research.)

8. Will any of the subjects be from one of the following vulnerable groups?:

- Children under 18 (16 in Scotland) Yes [ ]
- People with learning difficulties Yes [ ]
- Patients in hospital Yes [ ]
- Other vulnerable groups (e.g. mental illness, dementia) Yes [ ]

9. If any ‘yes’ box in question 8 is ticked, what special arrangements have been made to deal with issues of consent for the subjects (e.g. consent from parents, professional carer, relevant institution, etc).

10. Are there any potential physical, psychological or disclosure dangers that can be anticipated from involvement in the research?

Not exactly a psychological danger, but the application is designed as a resource for self-awareness, allowing users to re-experience, muse or get rid of thoughts and feelings they associate with events or people. They can occasionally experience unpleasant reflections.

If yes, please give details.

11. What steps will be taken to safeguard the confidentiality and anonymity of subjects? Identifiable data will not be stored, as information provided by subjects will keep anonymised and only used in dissertation’s evaluation.

12. Does the study design involve actively deceiving participants?

No.

If yes, briefly describe the nature of the deception and explain why it is necessary.

13. Does the research project comply with the requirements of current Data Protection legislation (for example, data storage and security.), including in relation to the use and (non-) disclosure of secondary data sets?

Yes.
14. Is your risk assessment of the health and safety implications for staff of the research

High / Medium / **Low or negligible**

*If medium or high please ensure that the health and safety officer in the school is informed.*

Please sign the following:

I as a Principal Investigator (supervisor) have checked the above for accuracy and am satisfied the information provided is a true reflection of the intended study.

Name (please print)  **LUIS ALBERTO GUTIERREZ IGLESIAS**

Signature

Date  **17/8/2016**

I am satisfied that the researcher has properly considered the ethical implications of the intended study and has taken appropriate action.

**PP** (Director of Research)

Date  **17/8/16**
Appendix E: Risk Assessment

Risk assessment should be made of all research projects. In our case, the risks are only those associated with office work.

**Risk Assessment form**

**MACS Risk Assessment Form (Project)**

**Student:**
Luis Alberto Gutierrez Iglesias

**Project Title:**
Diary Application: Microsoft Band Monitor

**Supervisor:**
Ruth Aylett

**Risks:**

<table>
<thead>
<tr>
<th>Risk</th>
<th>Present (give details) (tick if present)</th>
<th>Control Measures and/or Protection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Office environment-includes purely software projects</td>
<td>Present – The project consists on the development and evaluation of a mobile application</td>
<td>Nothing</td>
</tr>
<tr>
<td>Unusual peripherals e.g. Robot, VR helmet, haptic device, etc.</td>
<td>Not present</td>
<td></td>
</tr>
<tr>
<td>Unusual Output e.g. Laser, loud noises, flashing lights etc.</td>
<td>Not present</td>
<td></td>
</tr>
<tr>
<td>Other risks</td>
<td>Not present</td>
<td></td>
</tr>
</tbody>
</table>