CONTEXT-AWARE MOBILE DIARY

by

Hasan Faik Alan

B.S., Computer Engineering, Boğaziçi University, 2012

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering
Boğaziçi University
2014

ACKNOWLEDGEMENTS

First of all, I would like to thank my parents, Ayşe and Şefik, and my brothers, Alper and Prof. Yasin for their endless support and love.

I would like express my sincere gratitude to Prof. Bert Arnrich. I learnt a lot from him especially on setting realistic research goals and staying focused. I appreciate his help on the way of pursuing my academic goals.

I am grateful to Prof. Cem Ersoy for giving me motivation and guidance during my education in Boğaziçi University.

I would like to thank Prof. Özlem Durmaz İncel for being a member of my thesis committee.

I thank Gökçehan Kara, Orhan Ermiş, Hande Alemdar, Can Tunca, Bilgin Koşucu, Ali Akkaya, Rami Gökhan Kıcı, Yavuz Köroğlu and A. Murat Yagci who know the details of my research as much as I do through my countless presentations to them.

I thank Doctor Burcu Demiray for funding the data collection study and contributing to the improvement of our data collection system.

Finally, I would like to thank to the 22 participants of our data collection study for their genuine interest in my research and giving feedbacks even after the study.

This work was partially funded by the Co-Funded Brain Circulation Scheme Project "Pervasive Healthcare: Towards Computational Networked Life Science" (TÜBİTAK Co-Circ 2236, Grant agreement number: 112C005) supported by TÜBİTAK and EC FP7 Marie Curie Action COFUND.

ABSTRACT

CONTEXT-AWARE MOBILE DIARY

Human memory is fallible. As a remedy to this problem, in the near future wearable devices will have the potential to automatically record every bit of information concerning a person and his/her environment continuously. This way, lifelog of a person can be generated. In this thesis, we present a mobile lifelogging application that captures daily experiences of a smartphone user and allows to retrieve them later. In this direction, we first introduce a system named Smartphone Tracker which allows us to conduct large-scale data collection studies using smartphones. Using this system, we investigate the sensing capabilities of smartphones through a real life data collection study with 22 participants. Second, we present a novel algorithm which provides semantic location awareness to mobile devices. Using this algorithm, we are able to detect the entry/departure times to/from semantically meaningful places a user visits, such as home, office, parents' home, etc. Finally, we present a mobile lifelogging application named Auto Diary which automatically records SMS messages, phone calls, weather conditions, ambient audio and visited places. In addition, a retrieval functionality is offered which allows to retrieve past experiences via associative cues. For example, the cues from the statements "It was a rainy Monday morning. I was in my office and received a phone call from my brother between 10 am and 12 am." can be used as query terms in the application to retrieve the experiences (e.g., audio recordings) captured in the described day. We envision that Auto Diary can be useful for people experiencing episodic memory impairment. It can also be useful for people who want to remember their past experiences with greater precision.

ÖZET

BAĞLAM FARKINDA MOBİL GÜNLÜK

İnsan hafızası yanılabilir. Bu probleme bir çözüm olarak, yakın gelecekte giyilebilir cihazlar bir kişi ve kişinin çevresi ile ilgili her türlü bilgiyi otomatik ve aralıksız bir şekilde kaydetme potansiyeline sahip olacaklar. Bu sayede bir kişinin hayat kaydı oluşturulabilir. Bu tezde, bir akıllı telefon kullanıcısının günlük deneyimlerini kaydeden ve daha sonra geri elde edilmesini sağlayan bir mobil hayat-kayıt uygulaması sunulmaktadır. Bu bağlamda ilk olarak Smartphone Tracker adında, akıllı telefonlar kullanarak geniş çaplı veri toplama çalışmaları yürütmemizi sağlayan bir sistem sunuyoruz. Bu sistemi kullanarak, 22 katılımcılı bir veri toplama çalışması üzerinden akıllı telefonların çevresini algılama yeteneklerini inceliyoruz. İkinci olarak, mobil cihazlara anlamsal mekan farkındalığı sağlayan yeni bir algoritma sunuyoruz. Bu algoritmayı kullanarak, bir kullanıcının ziyaret ettiği ev, iş yeri ve ebeveynlerin evi gibi anlamsal olarak değerli mekanlara/mekanlardan giriş/çıkış zamanlarını belirleyebiliyoruz. Son olarak, telefon görüşmelerini, kısa mesajları, hava koşullarını, ortamın sesini ve ziyaret edilen mekanları otomatik olarak kayıt altına alan Auto Diary adında bir mobil hayat-kayıt uygulaması sunuyoruz. Ek olarak, geçmiş deneyimlerin ilişkisel işaretler ile geri getirilmesini sağlayan bir özellik de sunulmaktadır. Örneğin, “Yağmurlu bir pazartesi sabahıydı. Ofisimdeydim ve kardeşimden saat 10 ile 12 arası bir telefon aldım.” cümlelerindeki işaretler, tanımlanan gün içerisinde kaydedilen ses ve benzeri deneyimlerin geri getirilmesi amacı ile uygulama içerisinde sorgu terimleri olarak kullanılabilir. Auto Diary’nin epizodik bellek bozukluğu yaşayan insanlar için faydalı olabiliceğini öngörüyoruz. Uygulama, geçmiş deneyimlerini çok daha detaylı şekilde hatırlamak isteyen insanlar için de faydalı olabilir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

AP	Wireless Access Point
API	Application Programming Interface
BSSID	Basic Service Set Identification
CDD	Android Compatibility Definition Document
DT	Decision Tree
EMI	Experience Sampling Method
GCM	Google Cloud Messaging for Android
GPS	Global Positioning System
GSM	Global System for Mobile communications
KNN	K-nearest Neighbor
LAMP	Linux, Apache, MySQL, PHP stack
LBS	Location-based Services
MAC	Media Access Control
OS	Operating System
PDA	Personal Digital Assistant
SVM	Support Vector Machine

1. INTRODUCTION

Context-aware computing research aims to build systems that are able to sense an individual's changing context and adapt their operations accordingly [1]. There are numerous definitions of context in the literature [1–4]. Dey [4] defines context as

> *"any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves".*

Many forms of context-awareness can be found in a smartphone. Figure 1.1 shows an Android smartphone named Samsung Galaxy S5 produced by Samsung Electronics in 2014. It contains an accelerometer, a geomagnetic sensor, a gyroscope, a light sensor, a barometer, a proximity sensor, a gesture sensor, a fingerprint sensor, a hall sensor, and a heart rate sensor [5]. Most of these sensors can also be found in today's other off-the-shelf smartphone models. For example, the accelerometer and gyroscope are used in a smartphone to determine the motion of a smartphone. Using this information, in which orientation the user is holding the smartphone is determined and the user interface is adapted accordingly. The light sensor is used to measure the intensity of ambient light in an environment and the screen brightness is adjusted accordingly to optimize the screen visibility. The proximity sensor is used to disable the touch screen when making a call in order to prevent accidental touches of head to the screen. Fingerprint sensor, barometer, heart rate sensor and hall sensor are examples of new types of sensors in today's smartphones. Fingerprint sensor is used for biometric identification of the user. Barometer measures the atmospheric pressure and the altitude of the environment the user is in. Heart rate sensor measures the flow rate of blood cells in a user's finger and the heart rate of the user is determined. Hall sensor detects whether the smartphone cover which protects the screen is open or not.

Figure 1.1. Samsung Galaxy S5 smartphone. It contains ten different sensors: accelerometer, gyro, proximity, compass, barometer, hall, RGB ambient light, gesture, fingerprint and heart rate [5].

In addition to built-in context-awareness which mostly improves the usability of a smartphone, new context-aware applications can be developed for smartphones that make use of the available sensors. Mobile operating systems (OS) allow applications to access data of the sensors through application programming interfaces (API). For example, an application can estimate the type of physical activity that the user of a smartphone is performing such as walking, running, sitting and standing using the accelerometer [6]. Transportation mode of the user such as stationary, walking, running, biking and in motorized can be inferred using the GPS and accelerometer [7]. Microphone can be used to classify the ambient sound as music, speech or silence [8].

In this thesis, we utilize the context-awareness of smartphones to develop a mobile lifelogging application. Lifelogging is the process of recording information concerning a person and his/her environment continuously. Lifelogs can be a remedy to consequences of memory malfunctions [9, 10]. The invention of writing enabled human beings to encode their experiences in symbols and reconstruct them later in a systematic way. The invention of sensors such as microphone and camera enabled to store and retrieve our experiences with greater precision and efficiency. Smartphones equipped with various sensors are suitable platforms to develop lifelogging applications.

In this direction, we first introduce a system named Smartphone Tracker which enables us to conduct large-scale data collection studies using smartphones. Android, iOS, BlackBerry OS and Windows Mobile are some of the today's most used mobile operating systems. In 2013, 78.4% of the mobile devices sold worldwide use Android OS [11]. For this reason, we built Smartphone Tracker system targeting mobile devices running Android OS. There are many smartphone vendors such as Samsung, HTC, Sony and LG that produce mobile devices that run Android OS. Even though all of the mobile devices produced by these vendors comply with the requirements of Android OS [12], there are differences in the hardware components used in them such as the capacities of their batteries and the precisions of their sensors. Furthermore, there are different versions of Android OS with varying hardware requirements. In order to obtain an insight on smartphone usage patterns of users and the differences between different smartphone models, we conducted a five days long real life data collection study using Smartphone Tracker system. We present our insights from this study.

In order to determine semantically meaningful places of a smartphone user such as home, office and coffee shop, we developed an algorithm that relies on Wi-Fi infrastructure. In this algorithm, we periodically conduct Wi-Fi scans. When a wireless access point (AP) is found in consecutive Wi-Fi scans for a significant amount of time, the user is notified to give a name to his/her current location. When a name is supplied by the user, the algorithm associates nearby AP MAC addresses (BSSID) with the given name creating a Wi-Fi fingerprint for the location to remember the location in the future. To the best of our knowledge, there is no algorithm in the literature that employs real time user feedback to learn semantically meaningful places of a user.

Finally, we present a mobile lifelogging application named Auto Diary. This application logs phone calls, SMS messages, weather conditions, ambient audio and visited places. Visited places are logged using our novel algorithm. All of the captured experiences are shown in a daily timeline to the user. A retrieval functionality in this application allows to retrieve past experiences using associative cues. For example, a user remembering a rainy Monday morning in which s/he received a phone call from his/her mother can use these cues to retrieve his/her experiences (e.g., audio recordings) captured in the day described by the cues.

1.1. Contributions

Our contributions to the literature in this thesis can be summarized as follows:

- We present a system named Smartphone Tracker which allows us to conduct large-scale data collection studies using smartphones. We believe the design details of this system will be useful for other researchers working on smartphone sensing.
- We introduce a privacy preserving audio recording mechanism for smartphone data collection studies in which audio is transmitted to a server. In this mechanism, a smartphone user is able to listen automatically recorded audio files and choose which ones to upload to the server.
- We conducted a real life data collection study using Smartphone Tracker system with 22 participants for five days. In this study, we collaborated with a psychologist. We share our insights obtained from the study.
- We present a novel algorithm which provides semantic location awareness to mobile devices. This algorithm discovers semantically meaningful places of a mobile device user such as home, office and parents' home and tracks visits to the discovered places. We employ real time user feedback in this algorithm to learn both the names and the Wi-Fi fingerprints of the places.
- We present a mobile lifelogging application named Auto Diary. This application records phone calls, SMS messages, weather conditions, ambient audio and visits to places automatically. Visits to places are logged using our novel algorithm. All of the captured experiences are displayed in a daily timeline to the user. In addition, a retrieval functionality is offered in this application which allows the user to retrieve his/her past experiences using associative cues. We plan to put this application on the Android applications distribution platform.
- We developed a utility application named Sensor Log targeted for Android smartphones. This application eases the process of collecting and labeling sensor data from smartphones. The application can be downloaded from the Android applications distribution platform [1] . Details of this application are given in Appendix B.1.

[1] http://hfalan.com/sensorlog

1.2. Thesis Outline

The rest of the thesis is organized as follows: In Chapter 2, we present the related work in lifelogging, semantic location awareness, and smartphone sensing domains. In Chapter 3, we introduce a smartphone sensing system named Smartphone Tracker and give our insights from a real life data collection study which was conducted using the introduced system. In Chapter 4, we present a novel algorithm which provides semantic location awareness to mobile devices. In Chapter 5, we introduce a context-aware lifelogging application named Auto Diary. Finally, In Chapter 6, we conclude the thesis and discuss the future work.

2. RELATED WORK

2.1. Lifelogging

In 1945, Bush envisioned an electromechanical proto-hypertext machine that will allow individuals to store and share knowledge on microfilms [13]. The machine is described as a supplement to one's memory. In 2002, Microsoft Research presented a system named MyLifeBits as the fulfillment of the Bush's vision [14]. The first experiment was to archive digital life of Gordon Bell, an employee of Microsoft. Between 2001 and 2005, over 234,000 digital documents including emails, web pages, images, audio and video files belonging to Bell were archived [15]. In the experiment, a sensor augmented wearable camera named SenseCam, which was also developed by Microsoft, was used to capture photographs periodically [16]. The device is worn around the neck and takes photographs automatically [Figure 2.1]. It contains a light intensity sensor, a light color sensor, a passive infrared (body heat) detector, a temperature sensor and a multiple-axis accelerometer. The camera takes a photograph when it is triggered by a sensor (e.g., the detection of body heat or a change in the light intensity). The user can also set a timer to take photographs periodically.

Figure 2.1. SenseCam [17]. A sensor augmented wearable camera that captures photographs periodically or when it is triggered by a sensor.

Aizawa [18] presents a life log system that captures data continuously from a wearable camera, a microphone and various sensors all connected to a notebook PC as a main unit. System collects video, audio, acceleration, orientation, position from GPS, annotations, documents, visited web pages, emails and application usage information. From the collected data, keys that can be used for indexing the recorded video are extracted. For example, physical activity of the user such as walking, running or not moving is determined using accelerometer and gyro. From the GPS data, important location addresses, where a user spends significant time, are discovered. Various information from the Internet are recorded such as news and weather conditions. A user can also enter textual annotations at any time. All of the extracted keys can then be used to retrieve a video recording at a specific time described by the keys.

Abe *et al.* [19] proposes a lifelogging system to collect human behavior with higher level tags such as expenditure, tasks and accompanying-person. Data collection is performed on the fly using a smartphone application and post processing is done with a PC browser. A user can review the collected data such as locations on a map and assign tags using a PC browser. In addition, tags can be supplied on the smartphone application. When the smartphone application is started, tags such as location, purpose of being there, accompanying person are automatically guessed using the similarities from activity history and the user corrects the erroneous guesses. It is shown that the time spent on the logging application is decreased over time as the system captures behavior patterns of a user and finds out similarity among users.

Vemuri *et al.* [20] focuses on recording audio in daily life to alleviate long term memory problems. A software that runs on personal digital assistants (PDA) is designed to record audio when the user of the device activates it. The recorded audio files are processed using a third-party speech-recognition software to extract transcripts. Later, the user can retrieve relevant audio segments using the keywords that pass in the transcripts. A computer software is designed which displays the transcripts of recorded audio files. Each word in the transcripts is emphasized relative to recognition confidence given by the speech recognizer for it. A user study is conducted where the participants are given questions related to a three-day conference that was organized one month before the study. It is shown that using the designed audio retrieval mechanism, a user is able to find answers inside hours of audio recordings in a few minutes.

Frank *et al.* [21] uses probabilistic translation model to produce English-language descriptions of sensor events. The model is trained with English-language descriptions of events and corresponding sensor data. Hierarchical Dirichlet Process [22] is used for modeling and clustering Wi-Fi data. Stories similar to ”I left home at 04:20. I arrived at auditorium at 11:48. I arrived at home at 20:34” are produced.

Chennuru *et al.* [23] employ statistical natural language techniques to efficiently index, recognize, segment, cluster retrive and infer high-level meanings from lifelogs. A Nokia N95 phone mounted on a helmet is used as a lifelogger. Using this configuration, GPS coordinates, acceleration, sound, rotation sensor readings and Wi-Fi signal strengths are collected. Raw sensory data are quantized using K-means algorithm (KNN). Generated clusters constitute the “activity language” of the system. Similarity between different lifelog segments are calculated by measuring the similarity between corresponding activity language sentences. Using the web interface of the developed system, a user can select a lifelog segment and similar segments from the past are shown to him/her on the interface.

Blum *et al.* [24] developed a system named inSense which takes a picture and records a short audio clip when it detects an interesting moment. The system is a hardware platform which consists of a personal digital assistant (PDA) placed on the wearer’s chest and two wireless accelerometers one is placed left side of the hip and one is worn on the wrist. Location, speech, posture and activities are used to determine a user’s context. The system continuously records audio and acceleration signals, takes a picture every minute and logs Wi-Fi access points. Various machine learning methods are experimented to extract activities from the collected data. Later, a rule based approach is followed to determine interestingness of inferences from the data by assigning points to specific activity types. For example, laughter is given five interest points whereas no speech is given zero points. When the interest point exceeds a certain threshold, a picture is taken and audio is recorded. It is shown with a user study that when interest point based algorithm is employed system captures more interesting scenes compared to periodically taking pictures.

2.2. Semantic Location Awareness

Marmasse *et al.* [25] developed a mobile system to learn frequently visited locations of a mobile device user. The system uses GPS and employs user feedback. If GPS signal is lost in an area, then it is inferred that there is a building in the vicinity. A location from GPS is requested every ten seconds. If GPS signal is lost three times inside a given radius, the system marks the location as salient and the user is prompted to give a name to this location. The authors mention two of the main problems of their system. First, it is stated that the system experiences long delays acquiring GPS position especially when the user is exiting a building. In order to solve this problem, the system is modified to detect only arrivals. Second, GPS signal may sometimes be available inside some of the buildings which causes the GPS signal loss heuristic fail.

Ashbrook *et al.* [26] clusters GPS data to find meaningful places. GPS data are recorded with a wearable GPS receiver and a GPS data logger. The logger records GPS position with an interval of one second. It is mentioned that GPS signal is lost when the user enters a building and for most people buildings can be considered as significant places. A place is defined as any logged GPS point with an interval of time t between it an the previous point. The value of t, the minimum stopping time in a place in order to consider the place as significant, is taken as ten minutes. The collected GPS data are clustered with a variant of k-means clustering algorithm. In this algorithm, an initial GPS point and a radius are chosen. All the points in the radius of the chosen point are used to calculate a mean point. The mean point is then taken as the new initial point and the procedure is repeated until the mean point stops changing. Finally only the mean point is kept and other points are discarded. All of the GPS data are processed in this way to find significant places. A user study is conducted and after the data collection it is requested from the users to label significant places found in the data. It is found that common locations are named similarly by the users.

Montolu *et al.* [27] presents a framework to discover places of interest from multi-modal mobile phone data. A client-server architecture is developed to collect GSM, GPS, Bluetooth, Wi-Fi and accelerometer data from mobile phones. The client side application is installed to mobile phones. The application conducts Wi-Fi scans and learns the locations of the APs found in the scans by tagging them with GPS coordinates. First, the position of the phone is tried to be inferred from the nearby APs with a known location. Second, using the accelerometer and other sensors they determine the periods of time when a phone is static. When a phone is determined as static, the last known location is used. Finally, location is obtained from GPS if no nearby AP is found, the phone is not static or last location of the device is not known. Collected data are uploaded to a server. On the server side stay points which are clusters of location points and stay regions which are clusters of stay points with the same semantic meaning are determined. Three constraints are introduced in the algorithm to determine stay points. D_{max} is the maximum distance that a user can cover in a place, T_{min} is the minimum time that the user must be in the same place and T_{max} is the maximum time difference between two consecutive location points. They use $T_{min} =$ 30 min., $D_{max} = 250$ m. and $T_{max} = 10$ min. in the experiments. Density-based [28] and grid-based [29] clustering algorithms are experimented to determine semantically meaningful places defined as stay regions. Two experiments, one with eight people over five months and another with 124 volunteers along almost a year were conducted. The participants in these studies labeled the found places using 22 predefined labels. 92% of the places found by the algorithm were known by the participants. The algorithm determined the homes and the work places of the participants with 97% and 79% accuracies respectively.

Kang *et al.* [30] extracts important places of a user from the locations obtained by Place Lab [31], a coordinate-based location system that keeps locations of APs in a database. The main idea of their algorithm is to cluster the locations in the time domain. Clusters are formed incrementally with the acquisition of each location measurement. Time threshold (t) and distance threshold (d) are the two parameters of the algorithm. If the distance between the mean point of an existing cluster and a new location is less than d, the location is added to the cluster. If the distance is larger than d, a new cluster is formed with the location. If the time spent in a cluster is more than the time threshold t, a new significant place is found. d determines the size of clusters. t determines the number of clusters. In the experiments, d is chosen between 30m and 50m. t is chosen at least 300 seconds. They show that the algorithm is able to discover clusters in location traces where conventional clustering algorithms such as k-means and Gaussian mixture model fail.

Ye *et al.* [32] determines significant places of an individual based on spatial and temporal analysis of GPS data. If an individual spends more than 30 minutes within a 200 meters range then this location is determined as a stay point. Stay points are then clustered using a density based clustering algorithm and visits to these clusters of stay points are used as the location history of the individual.

LaMarca *et al.* [33] introduce a radio beacon based approach called Place Lab to determine location of a mobile device. Wi-Fi access points, GSM cell towers and Bluetooth devices are called as beacons. Place Lab keeps a database of unique beacon identifiers mapped to physical locations. For example, APs and bluetooth devices broadcast their MAC addresses to be found by other mobile devices. GSM cell towers have unique cell identifiers. Place Lab uses a venn-diagram-like intersection of observed beacons to estimate the location of a mobile device. In the experiments it is shown that when there is sufficient density, AP beacons provide median accuracy of 20 meters whereas GSM beacons provide 100-200 meters accuracy. It is mentioned that Bluetooth scans take approximately ten seconds to complete and even with walking speed a mobile device misses Bluetooth beacons. Furthermore, Bluetooth device density is found very low in many environments. However, in an experiment it is shown that when Bluetooth beacons are deployed to an environment strategically they increase the location estimation of Place Lab.

Laasonen *et al.* [34] uses GSM network cell information to discover important places of a person without knowing the physical topology of the network. Timestamped sequence of GSM cell identifiers constitute the location data and transitions between different cells are determined. All of the cells with a transition time below a certain threshold between them are put into same cluster. $t_1 = 3$ min, $t_2 = 10$ min and $t_3 = 60$ min are used to cluster cells in different time transition resolutions. t_1 captures towns or cities, t_2 discovers individual cities and t_3 discovers regions of countries. However, many problems are mentioned with their system. Cells can be large up to kilometers in diameter. There may be overlaps between several cells and due to changing radio interference one-to-one correspondence between a physical location and a cell is not possible.

Hightower *et al.* [35] present an algorithm named BeaconPrint which periodically scans for Wi-Fi and GSM signal sources called beacons. The algorithm uses a time window w. In w, all of the unique beacons nearby are determined and scan count histograms of the beacons are generated. After w, if there is no new beacon around, the algorithm infers that the radio environment is stable. When the mobile device enters a new environment, it compares the live fingerprint with the beacon histograms of previously learned places to determine most similar fingerprint. Performance evaluation of the algorithm is done using three people's montly GPS traces. BeaconPrint achieved over 90% accuracy in learning and recognizing places.

Kim *et al.* [36] presents an algorithm named PlaceSense to discover semantically meaningful places from Wi-Fi or Cell tower fingerprints. The algorithm is similar to BeaconPrint algorithm which is introduced in [35]. It uses a scan window to determine beacons in the environment. A scan window is determined as stable if it does not contain any new beacons. The number of stable windows required to determine an entrance to a place is given as a parameter. A response rate which equals to ratio of detection count of a beacon to total number of scans in a window is calculated for each beacon in the window. Beacons with a response rate greater than a certain threshold are determined as representative beacons of a place. A departure is determined if all of the representative beacons of the current place disappear and new beacons are found in a scan window. In the experiments window length is used as two minutes. Number of stable windows required to determine an entrance is used as three. It is shown that PlaceSense discovers 92% of the visited places and detects entrance and departure times.

Kim *et al.* [37] presents a semantic location service named Loci which runs on a mobile device and uses Wi-Fi fingerprints as input. It learns new places, suggests potentially meaningful places, recognizes registered places and tracks paths connecting places. The service gathers potentially meaningful places during the day and the user reviews them. A new place is determined when the user stays more than five minutes in a place. A Wi-Fi scan is conducted every ten seconds. If Wi-Fi fingerprint similarity is above a certain threshold for three consecutive scans, an entry to a place is determined. Accelerometer is used to determine times when the device is immobile and Wi-Fi scans are stopped if the device is not moving. A mobile user interface is designed in which a user can review the suggested places. A user can add a suggested place to an existing place which allows to associate several Wi-Fi fingerprints with a place. The locations of Wi-Fi access points are learned using GPS. The locations of APs, Wi-Fi access point names and recent visit times are shown to user in order to assist the user determining valid places among the suggested places. It is mentioned that in order to improve semantic location services, user feedback with well-timed prompting mechanisms are needed. In addition, in order to improve the algorithm's performance incorporating the user feedback into the place detection algorithm as a parameter is included in the author's research agenda.

2.3. Smartphone Sensing

Froehlich *et al.* [38] presents a system named MyExperience which runs on Windows Mobile devices and captures mobile computing activities of a user through automatic data logging and event-driven experience sampling. The system supports logging more than 140 event types related to phone calls, SMS, application usage (e.g, music, video and games), calendar appointments, contacts, Bluetooth, GSM and Wi-Fi. It offers an XML interface for researchers to define survey questions and configure sensors. Through embedded scripts, conditions can be defined to trigger surveys. For example, a script can be defined such that when a phone call is finished with 20% probability a survey is displayed to the user. Using this system, several studies are conducted such as battery charging behavior analysis and SMS usage analysis. In SMS usage study, a survey is triggered after each SMS is sent. Questions investigating the participant's location, the purpose of the sent message and the reasons for using SMS over other communication technologies are included in the survey. The reasons mentioned for SMS use were "responding to a previously received message", "convenience/faster" and "couldn't use voice". In the battery charging behavior analysis study, the users are surveyed at the moment when they start charging their phone. The reasons mentioned for charging the phone were "to synchronize the phone", "battery was getting low" and "habit".

Aharony *et al.* [39] presents an open source smartphone sensing framework named Funf targeted for Android smartphones. Various types of data can be collected periodically using this framework from smartphones including sensor data, battery level, location from GPS and application usage. The framework can be extended to collect any type of data that Android OS allows to access. In this thesis, we use Funf framework as a part of our smartphone sensing system named Smartphone Tracker which is introduced in Section 3.1. In Section 3.1.1, we give a detailed description of Funf framework.

Miluzzo *et al.* [40] developed a mobile application named CenceMe which infers the context of a smartphone user and shares the inferred information through social networking applications such as Facebook and MySpace. The application is targeted for Nokia N95 mobile phones. Data from accelerometer, GPS, microphone and Bluetooth are used to make inferences on the mobile phone. Sound samples from microphone are used to classify the ambient sound as human voice, noise or silence. Accelerometer is used to determine the type of physical activity performed such as walking, standing and running. GPS speed is used to determine the mobility of the user as stationary, walking or running. GPS location history is processed to discover significant places and visits to these places are logged. MAC addresses found in Bluetooth scans are used to determine nearby people who also use CenceMe application. Collected data are transmitted to server and high level inferences are made on the server side. For example if sound level is above a certain threshold and activity is close to running, the user is determined as dancing. Facebook status of a user is then updated using the inferred activities such as "Joe is at work, in a conversation, standing".

Wu *et al.* [41] presents a mobile sensing platform named MobiSens which consists of an Android smartphone application and a backend system. Using this system various types of information can be collected from smartphones such as data from sensors, GPS coordinates, nearby Bluetooth devices, calendar entries, charging state, ringtone settings and running processes. The system performs activity recognition on the client side and recognized activities are shown to the user. A user is able to correct erroneously recognized activities. Collected data are also uploaded to server and stored on the server. Three different versions of MobiSens Android mobile application are released to the Android Market. Effect of UI design changes which motivate the users and the sensing performance of the application are investigated with the each release of the application. 13,993 hours of data are collected from 310 users in over five months. Through design changes which motivate the users, average number of activity annotations supplied by a user increased from 0.6 to 6.

Lane *et al.* [42] presents an Android mobile application named BeWell which tracks the user's sleep pattern, physical activity and social interactions. The accelerometer data are used to classify the user's physical activity as driving, stationary, running or walking. Phone charging events, times when the phone is stationary or in a silent environment are used as features to determine when the user is sleeping. Social interactions are determined using the microphone by looking for human voice in the captured audio. The application calculates scores for each of the three dimensions and gives feedback to the user in order to promote better health.

Montjoye *et al.* [43] uses the text and call logs of a mobile phone to predict the personality of the user. The data from 69 participants' Android smartphones are used in the study. The data are collected using the Funf open sensing framework [39]. Psychology-informed metrics namely basic phone use, active user behavior, mobility, regularity and diversity are introduced. 36 different indicators each belonging to one of these metrics are used. For example, number of calls made is used as an indicator of active user behavior. Number of places where calls are made from is used for mobility metric. Number of interactions by number of contacts ratio is used for diversity metric. Using these indicators they determine how extroverted, agreeable, conscientious, open to experience, and emotionally stable the user is. They achieve 42% accuracy using Support Vector Machine (SVM) classifier in determining the personality of the user.

LiKamWa *et al.* [44] developed a smartphone service that can infer the mood of the user based on the information available in the smartphone. They conducted a field study with 25 iPhone users. LiveLab iPhone Logger [45] is used to collect application usage, phone calls, emails, SMS messages, web browsing history, calendar entries and location information from the smartphones of the users. User interfaces are designed where the users can input their mood in happy-sad and aroused-sleepy dimensions. Various histograms are produced from the collected smartphone usage data. These histograms are then correlated with the ground truths obtained from the user mood inputs. It is shown that the mood of a smartphone user can be inferred with 91% accuracy.

Moturu *et al.* [46] investigated the associations between sleep, mood and sociability. A study with 54 participants is conducted. Each of the participants use an Andorid smartphone with Funf smartphone sensing [39] framework installed. Sleep duration and mood information are self reported by the participants in periodic surveys. Funf framework is configured to scan nearby Bluetooth devices every five minutes. Face-to-face social interactions are inferred from presence of nearby Bluetooth devices. Significant bidirectional relationship between sleep and mood is found. People with lower sociability reported poor mood. Higher sociability is observed when the night's sleep is between 7-8 hour range.

Wang *et al.* [47] developed an energy efficient mobile sensing system. Sensors are opened and closed (duty cycles) by guessing user state and activity transitions. Speed value obtained from GPS is used to infer the mode of travel. Wi-Fi is used to learn frequently visited places and movement ranges in those places. When a user is detected moving via accelerometer, a Wi-Fi scan is conducted to determine movement range. If GPS signal is lost in a location, the location is determined as indoor. A user trial is conducted where users recorded their current state using labels from transportation mode (e.g., walking, running, in vehicle, etc.), location and background sound (e.g., quiet, loud, speech etc.) categories. It is shown that battery life of a phone is inreased over 75% while maintaining high accuracy in activity recognition compared to state of the art activity recognition systems.

Do *et al.* [48] shows strong dependencies betwen phone usage and two contextual variables namely places and social context. A client-server architecture is used to collect data from the smartphones of 77 people for over nine months. The client side is an application that runs on Nokia N95 phones. Various types of data are collected from the phones including GPS, GSM, Wi-Fi, accelerometer and usage of apps. A statistically significant correlation is shown between app usage and the users' locations. For example, while SMS is used mostly in indoor locations, people prefer to use voice call in moving contexts such as in a metro station or while shopping. Clock application is used mostly at home and a friend's home. Significant correlations are also found between social context and app usages. Number of devices found in the Bluetooth scans is used as a measure of human density nearby and as an indicator of the social context of the users. It is found that when the Bluetooth density is high, there is an increase in SMS and voice call usage. It is also found that Clock application is used mostly at times when the Bluetooth device density is low such as when the users are at home.

Falaki *et al.* [49] investigates the smartphone usage traces of 33 Android users and 222 Windows Mobile users. Smartphone usage of the participants are analyzed in four dimensions namely user interactions, application use, network traffic and enery drain. Significant diversity in smartphone usage among the participants is found. For example, the number of interactions with the smartphone varies between 10 to 200. The number of applications used varies between 10 to 90. The mean network traffic per day varies between 1 to 1000 MB. Diurnal patterns are observed in each of these dimensions. For example, messaging application is found to be popular during the day than night. A personalized energy drain predictor is designed based on user specific smartphone usage patterns. Results suggest that user experience design and energy optimization mechanisms should adapt to user behavior instead of using generic models.

Oliver [50] analyzes smartphone interaction traces of 17300 BlackBerry users. Collected data are analyzed using similar dimensions that are used in [49]. It is found that average daily interaction time of the users is 1.68 hours and 80% of the interactions are less than 90 seconds in length. Diurnal patterns similar to [49] are found. For example, a strong correlation between hour of day and user interaction is found. Challenges of smartphone sensing are discussed. A malicious user may alter the locally collected data on a smartphone which may affect the quality of a study. Third-party applications such as battery optimization applications may interfere with the data collection application of a smartphone sensing study.

Lu *et al.* [51] developed a continuous sensing engine named Jigsaw without sacrificing accuracy in activity recognition and increasing energy consumption. Jigsaw uses a set of pipelines for accelerometer, microphone and GPS sensors. The system adapts to the quality of the sensor data and the mobility of the smartphone user. For example, GPS sampling rate is adapted to the user's mobility pattern. Duty cycle of microphone is adjusted to acoustic events recognized. The system is designed in an extensible way that new sensors and pipelines can be added.

3. SMARTPHONE SENSING

In this chapter, we first introduce a smartphone sensing system named Smartphone Tracker. We developed this system in order to conduct large scale data collection studies using smartphones. Later, we present our insights from a five days long data collection study which was conducted using the Smartphone Tracker system. In this study, we collected various types of data from 22 people's smartphones.

3.1. Smartphone Tracker System

In this section, we introduce Smartphone Tracker system which enables us to conduct large-scale data collection studies using smartphones. The system consists of three main components namely a dashboard, a server and an Android application. The architecture of the system is given in Figure 3.1.

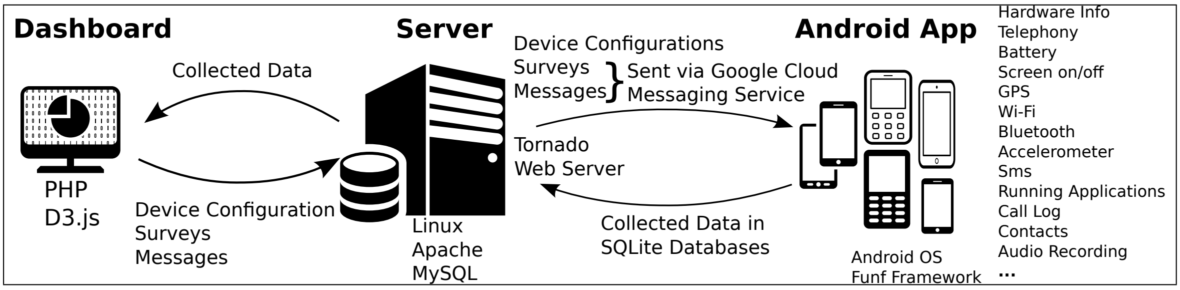


Figure 3.1. Smartphone Tracker system architecture.

The Android application uses the MIT Media Lab's open source smartphone sensing framework named Funf [39]. Funf collects various types of information from a smartphone such as battery level, running applications and nearby bluetooth devices. Collected data are periodically uploaded to the server. On the server side various web technologies are utilized such as Tornado Web Server [2] to facilitate communication with smartphones; LAMP (Linux, Apache, MySQL, PHP) stack for data storage and the dashboard; and d3.js [3] for visualizations on the dashboard. The dashboard allows

[2]http://www.tornadoweb.org/
[3]http://d3js.org/

to configure the data collection application installed on the smartphones and visualize the collected data. We integrated a survey infrastructure to the system. Surveys are created on the dashboard and sent to the smartphones via Google Cloud Messaging service (GCM) [4] . We also implemented a messaging infrastructure utilizing GCM service to send messages to the smartphones when we need to communicate with the users of the smartphones. In the following sections, these components are described in detail.

3.1.1. Funf Framework

We use opensource Funf framework as a part of our Android application [39]. This framework allows to collect various types of information from an Android smartphone such as running Android OS version, battery level, nearby Wi-Fi and bluetooth devices, location from GPS, running applications, installed applications, browser searches, call logs and data from sensors available on the device. In the framework's terminology, each of these signals are called as a probe. Various types of Funf sensing modalities and probes under these modalities are illustrated in Figure 3.2. For each of these probes period and duration of data collection, when applicable, can be specified in a configuration file. For example, accelerometer data can be collected every minute for ten seconds. Funf also offers probes that process collected raw data before storing them. For example, activity probe infers the physical activity level of a smartphone user by thresholding the variance of acceleration magnitude values as high, low or none. The framework also allows to add new probes and replace the existing ones with new implementations. This basically means it can be extended to collect any information from a smartphone that Android OS allows to access. All of the collected data are stored in SQLite databases. Sensitive data such as phone numbers used in calls and text used in browser searches are one-way-hashed before storing them in the database. Locally stored databases are periodically uploaded to a server with a specified URL over HTTP protocol. Data collection parameters can be changed remotely. Funf periodically controls a specified URL for an updated configuration. Data upload and update control periods are also specified in the configuration file.

[4] http://developer.android.com/google/gcm/

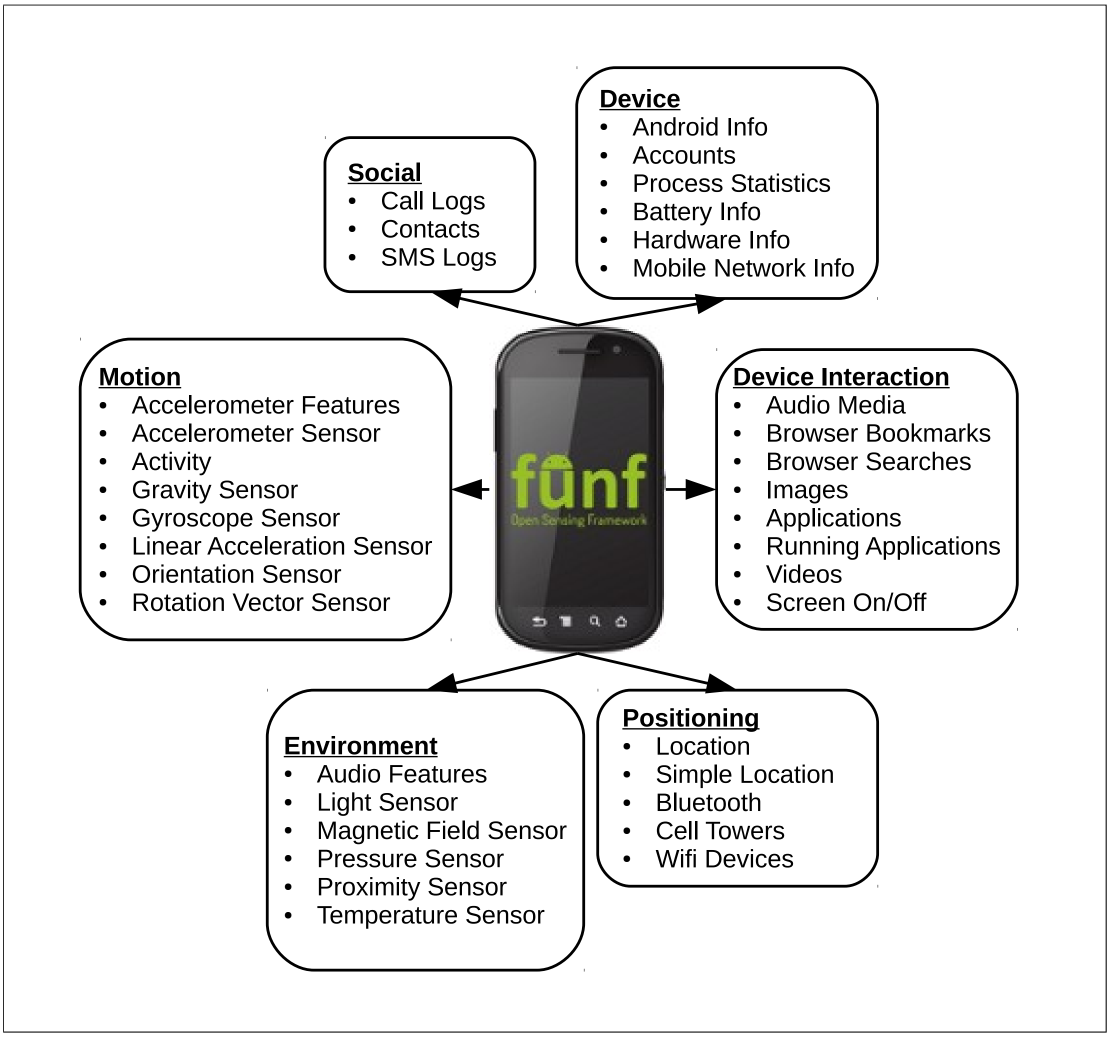


Figure 3.2. Funf framework data modalities and probes.

3.1.2. Android Application

We developed an Android application named Survey which constitutes the client side of Smartphone Tracker system. This application is installed to smartphones that we want to collect data from. We use Funf which is described in Section 3.1.1 as a framework in our Android application. Using the application, we can also conduct surveys. During the day, survey notifications which request a survey to be filled are sent to smartphones. We can also communicate with the users of the smartphones by sending messages using GCM. When audio recording is configured, Funf framework records audio periodically using the microphone of a smartphone. However, privacy of the smartphone users are violated if the raw audio is directly transmitted to server without their knowledge. Thus, we created user interfaces which allow a smartphone user to review recorded audio files before they are uploaded to server. Using these interfaces a user listens the recorded audio files and decides which ones to be used in our studies. The details of these mechanisms are given in the following sections.

3.1.2.1. Survey Mechanism. Surveys are created on the dashboard. Survey related information such as questions and choices are then sent to smartphones using GCM service. Periodic survey notifications are scheduled on the Android application. When a notification is received, a smartphone beeps and vibrates. Survey notifications are shown on the notification area of a smartphone [Figure 3.3]. When the notification is tapped Survey application is started.

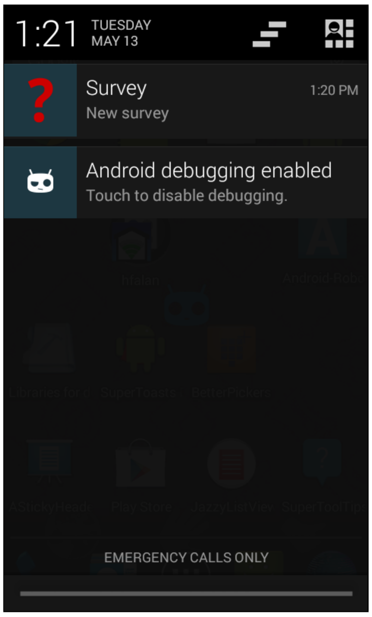


Figure 3.3. Screenshot of the notification area when a new notification is received.

There are 4 different question types in Survey application. These are multiple choice multiple answer, multiple choice single answer, audio recording and rating. Screenshots of 4 sample questions each belonging to one of the available question types are given in Figure 3.4.

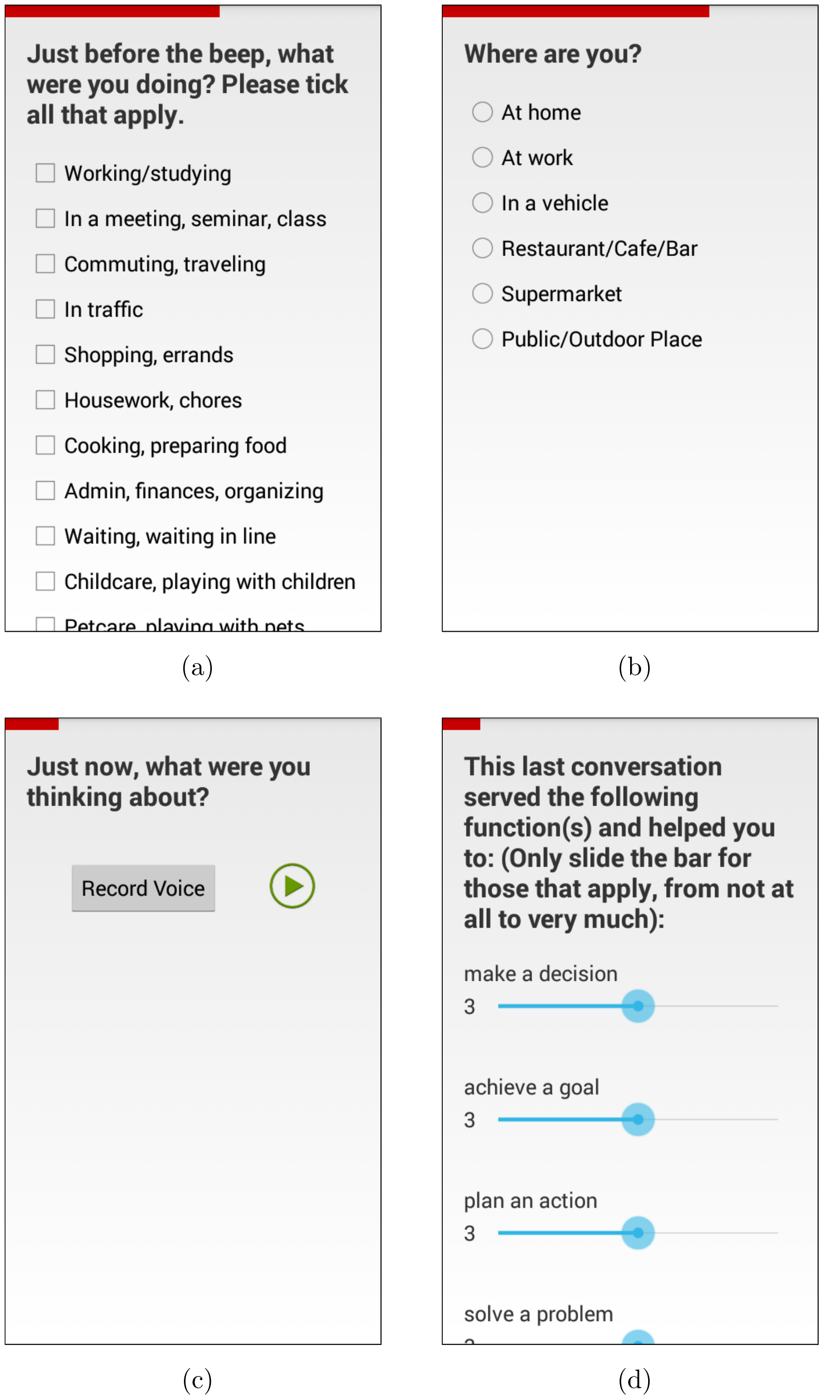


Figure 3.4. Screenshots of 4 sample questions in Survey application: (a) multiple choice multiple answer (b) multiple choice single answer (c) voice recording (d) rating.

3.1.2.2. Messages Interface. We are able to send messages to smartphones from dashboard using GCM service. Messages sent from dashboard can be seen in the messages section of Survey application [Figure 3.5]. Similar to a new survey notification, a notification is sent when a new message is received. When the user taps on the message notification, Survey application is started and messages section is shown in the application. We use the messaging infrastructure when we need to communicate with the user of a smartphone. For example, Android does not allow an application to enable GPS of a smartphone without the user's approval. Using the messaging infrastructure, we can send a message to a user that requests enabling the GPS service of the device.

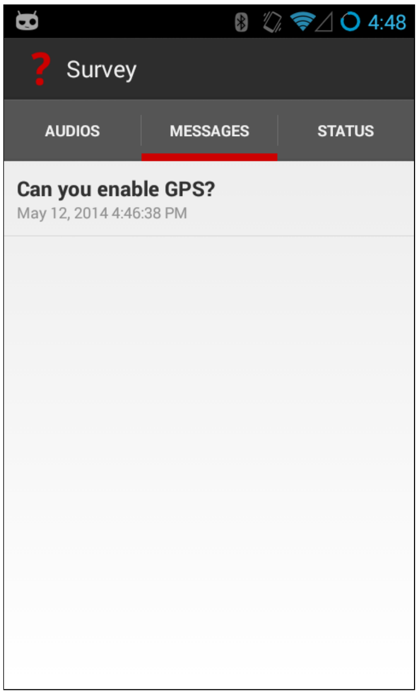
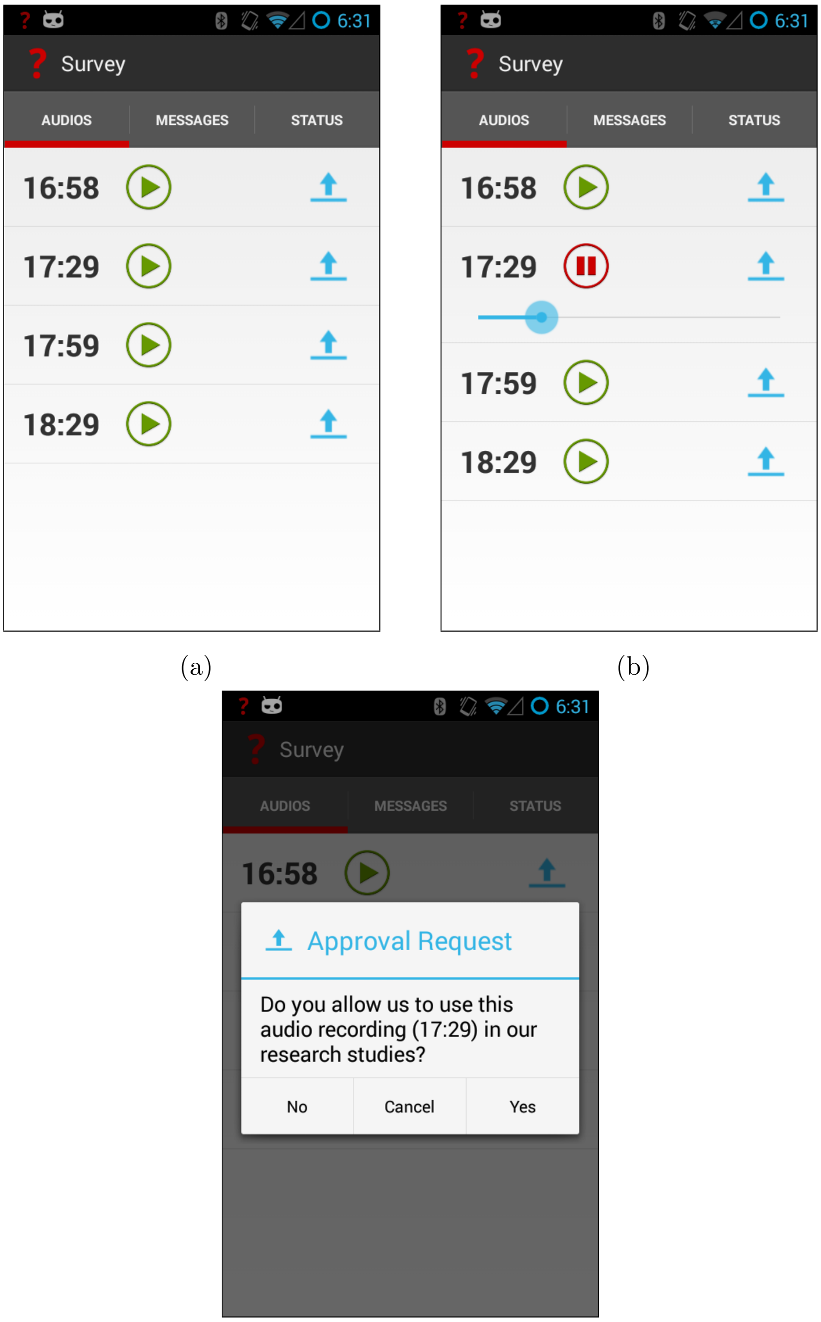


Figure 3.5. Screenshot of messages interface in Survey application.

3.1.2.3. Privacy Preserving Audio Recording Mechanism. When the audio recording probe is configured, Survey app records audio periodically in the background. Normally, Funf framework uploads the recorded audio files to server automatically. We created audio related interfaces in Survey app in order to allow the user of a smartphone to review recorded audio files before they are uploaded to server. These interfaces can be seen in Figure 3.6. Audio recordings are listed under the audios section (Figure

3.6a. When a play button is tapped, corresponding audio file starts to play (Figure 3.6b). When the audio playing is finished, user approval is requested for uploading the audio file to the server (Figure 3.6c). If the user says yes, the audio file is archived for uploading it to the server in the next upload cycle. If the user says no, the audio file is deleted immediately. This way we preserve the privacy of the users.



(a) (b)

(c)

Figure 3.6. Screenshots of audio related interfaces in Survey Application: (a) audio list (b) audio is playing (c) user approval is requested to upload audio to server.

3.1.3. Dashboard

When Survey application is installed on a smartphone, it first generates a unique ID for the smartphone. Together with the hardware information of the smartphone this unique ID is sent to server. In the dashboard all of the smartphones which have sent their unique ID are displayed under the list of devices section [Figure 3.7]. In this section brand, model, unique ID, join time and last access time of each device can be seen. In the detail screen of each device latest known state of the device is shown [Figure 3.8]. Each device has a configuration panel and data collection parameters can be set per device basis [Figure 3.9]. Also a default configuration can be applied to all of the devices connected to the system. Data collected from battery, location, screen state, accelerometer and application usage probes are visualized on the dashboard [Figure 3.10 - 3.14]. Visualizations helps us to discover problems related to the smartphones and get insights from the collected data. For battery level and screen state probes last 48 hours of data are visualized. A heatmap is generated for location history [Figure 3.11]. Latest known location of the device is also shown with a marker. Application usage is visualized with a pie chart that displays total usage time distribution among applications [Figure 3.14]. We use GCM service to send messages to smartphones. If a smartphone is not connected to Internet when a message is sent to it, it receives the message immediately the first time it connects to Internet. New surveys are also created from the dashboard. Multiple choice multiple answer, multiple choice single answer, rating or audio recording can be chosen as a question type while creating a new question. We also send messages to smartphones from the dashboard.

3.1.4. Server

Tornado server is used to communicate with smartphones. Smartphones periodically send their SQLite databases to Tornado server over HTTP protocol. Tornado server extracts the data stored in SQLite files and imports them into MySQL database after each HTTP post request in order to make them available for visualization. SQLite databases are also stored on the server side for offline analysis of the collected data.

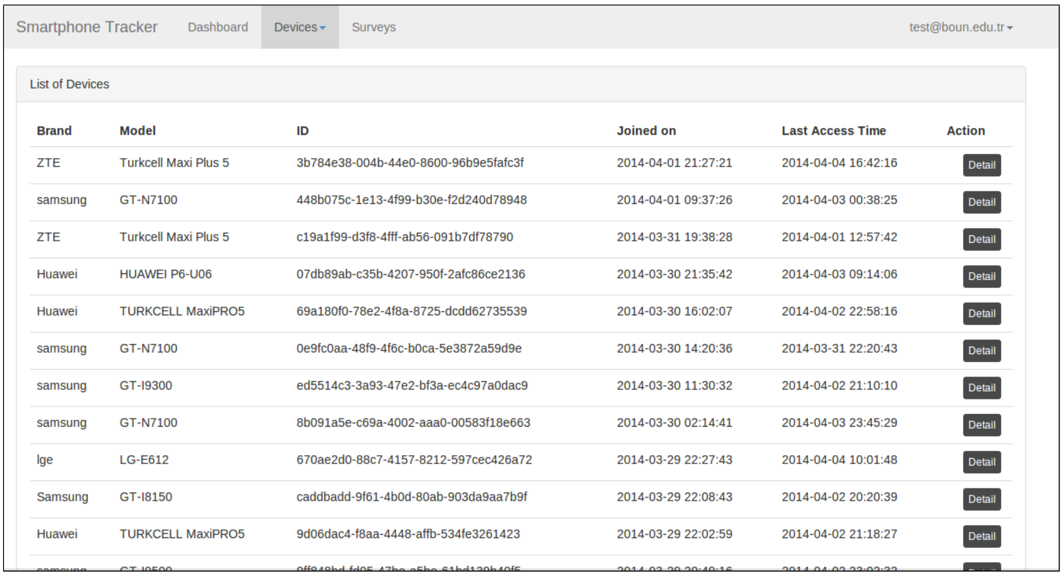


Figure 3.7. Smartphone Tracker user interface for reviewing connected devices.

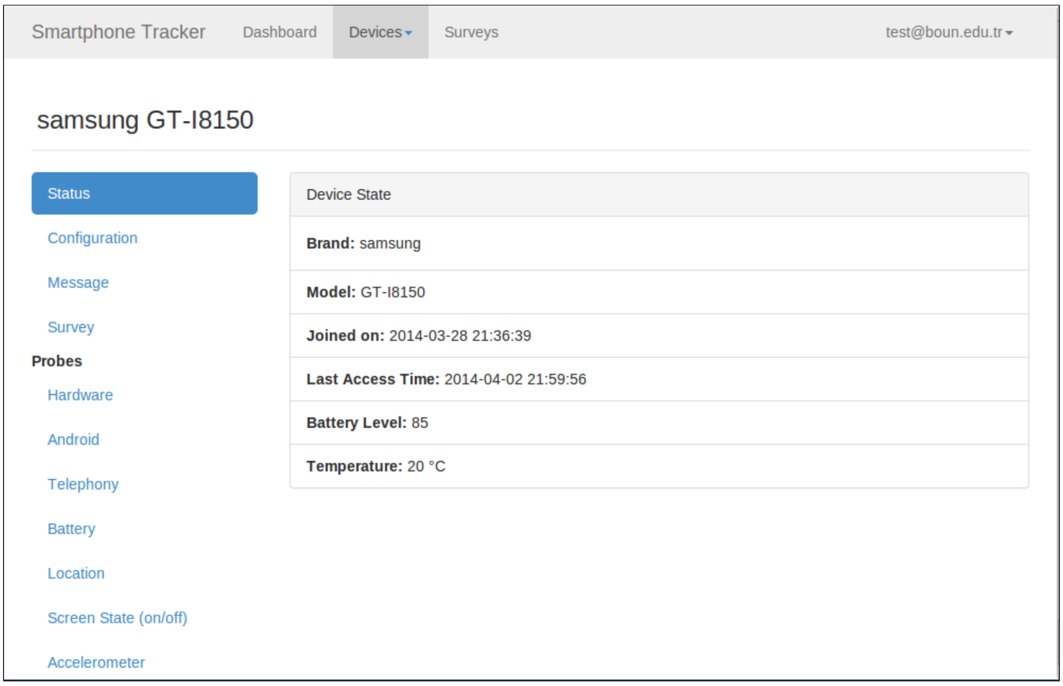


Figure 3.8. Smartphone Tracker user interface for device status review.

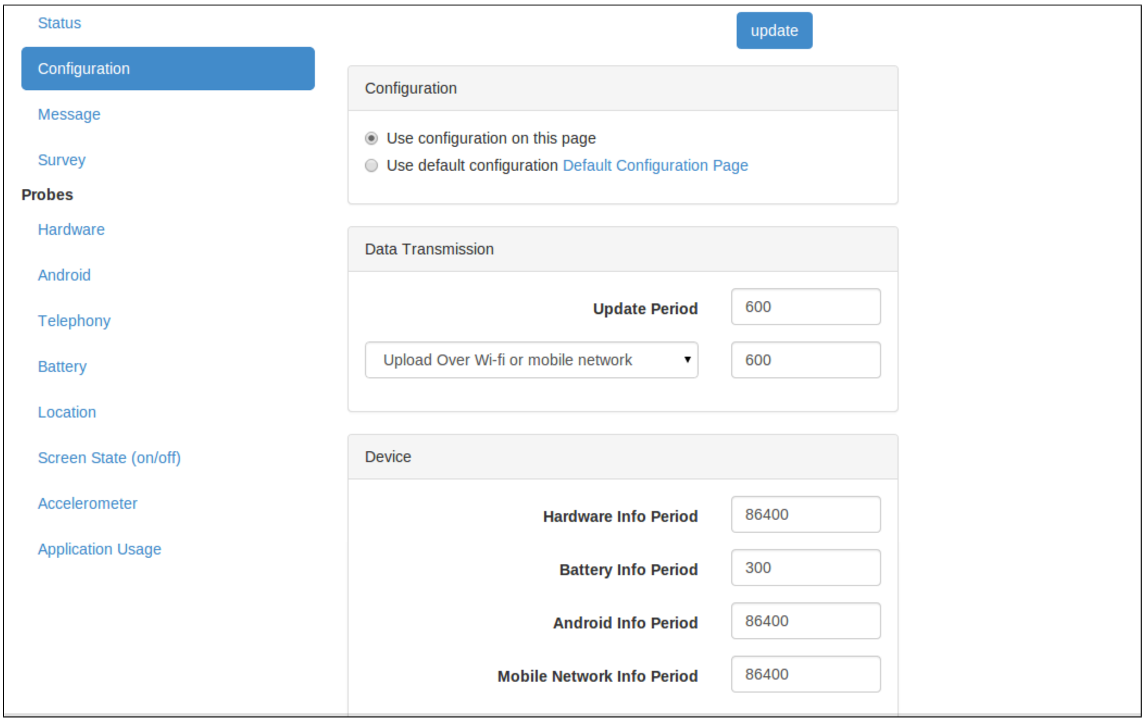


Figure 3.9. Smartphone Tracker user interface for device configuration.

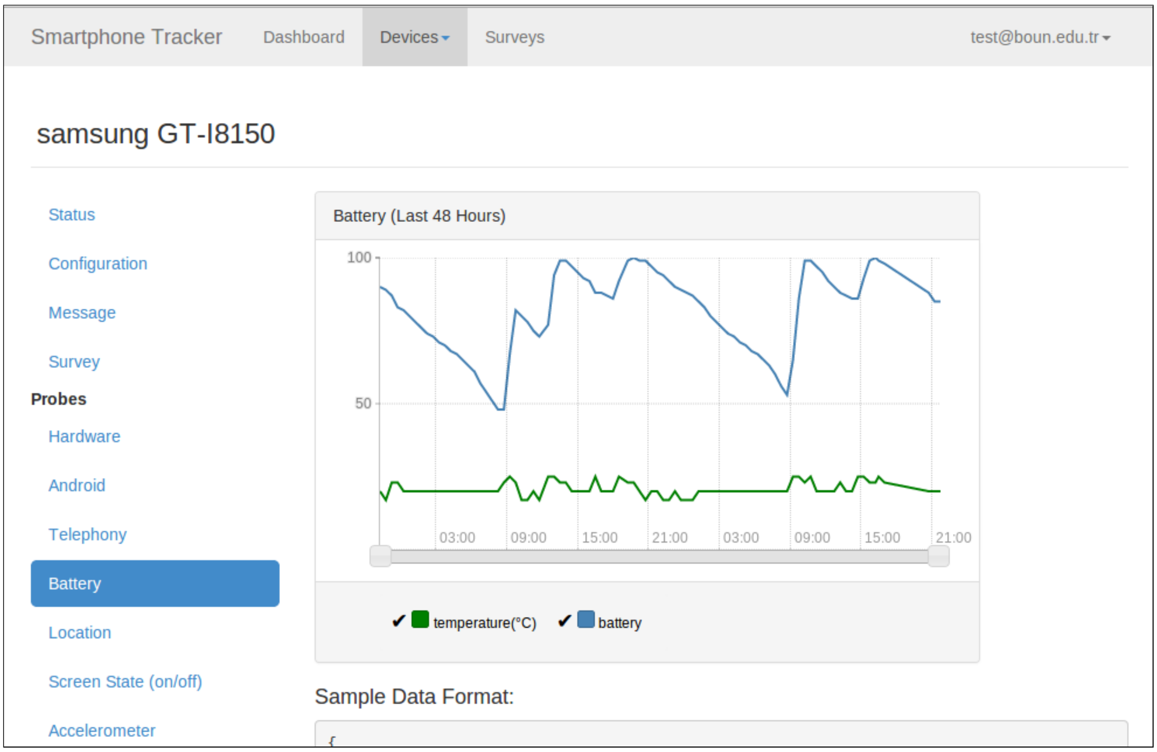


Figure 3.10. Smartphone Tracker user interface for battery data visualization.

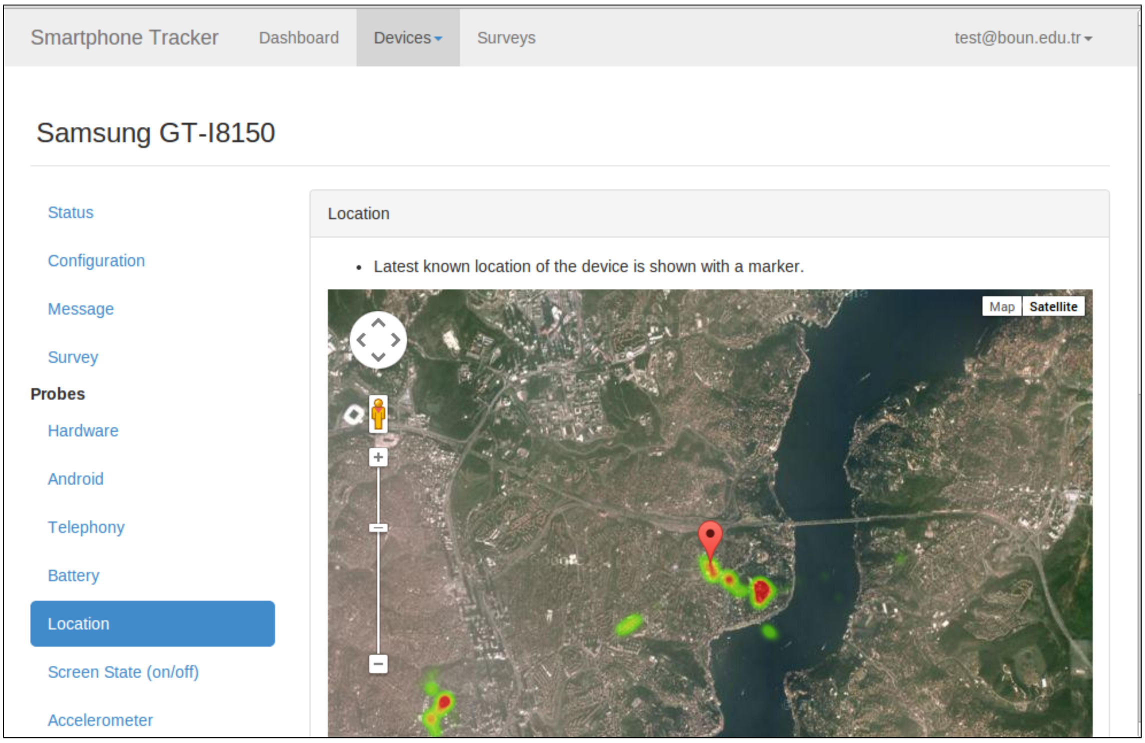


Figure 3.11. Smartphone Tracker user interface for location data visualization.

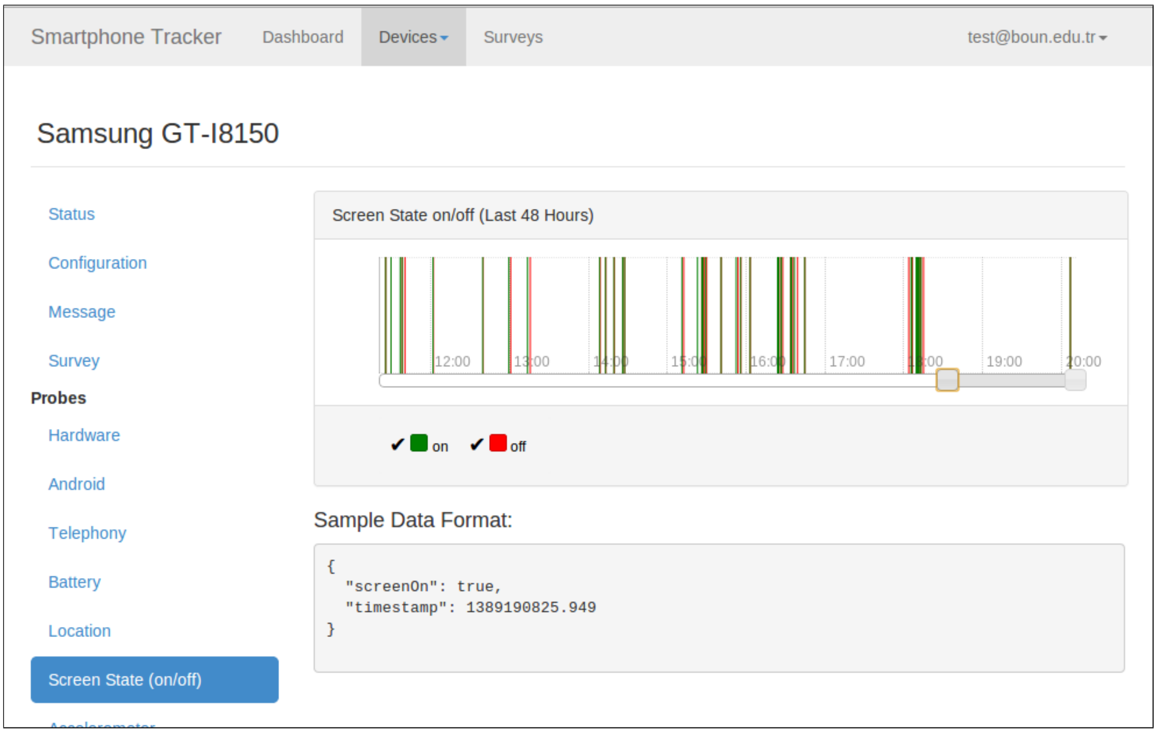


Figure 3.12. Smartphone Tracker user interface for screen on/off data visualization.

Figure 3.13. Smartphone Tracker user interface for accelerometer data visualization.

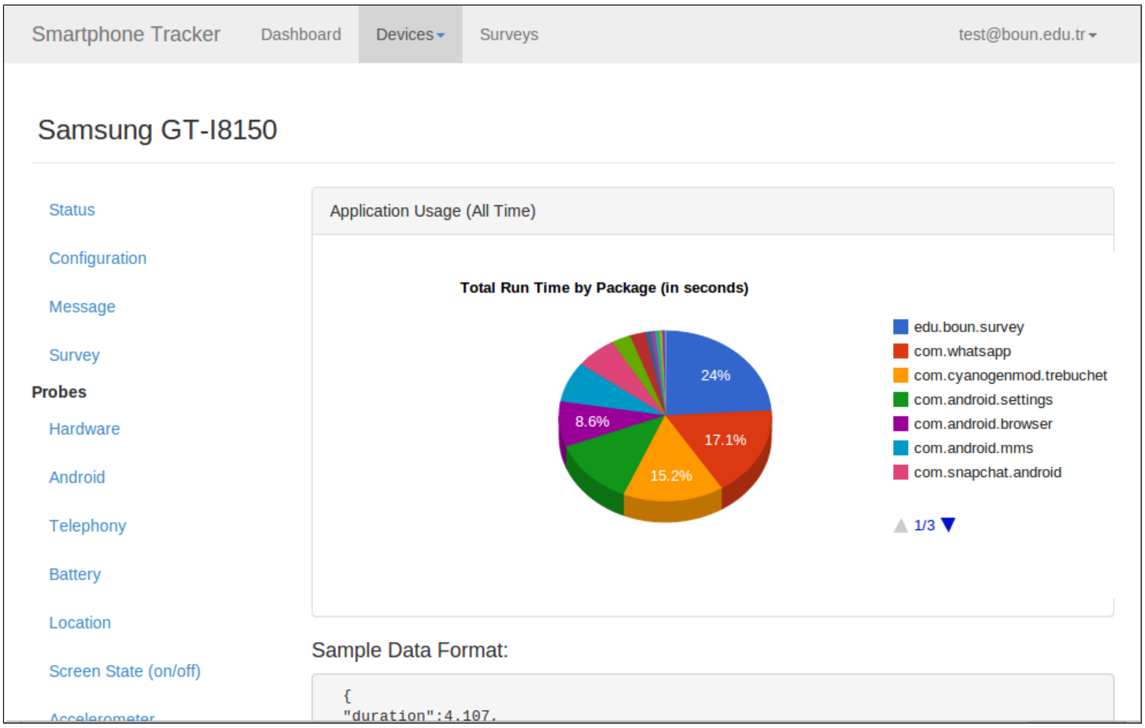


Figure 3.14. Smartphone Tracker user interface for application usage visualization.

3.2. 5 Days 22 People Data Collection Study

In this section, we present our insights obtained from a data collection study conducted using Smartphone Tracker system which is introduced in Section 3.1. In this study, we collaborated with a researcher from the Department of Psychology at University of Zurich.

3.2.1. Motivation

The ease of developing and distributing smartphone applications that can sense their environment gives an unprecedented opportunity to researchers from various disciplines especially the ones from social sciences. A researcher from the Department of Psychology at University of Zurich informed us that she want to conduct a study using smartphones in order to get a comprehensive picture of the smartphone users' daily activities, social interactions and personal thoughts. According to the requirements of this study, we improved Smartphone Tracker system in two ways. First, to be able to capture personal thoughts of the users via experience sampling method (EMI) [52], we created the survey mechanism as described in Section 3.1.2.1. Second, the collaborator wanted to have real-life narratives of the smartphone users. In order to satisfy this requirement, we implemented the privacy preserving audio recording mechanism which is described in Section 3.1.2.3. In the following sections we give the details of the conducted study.

3.2.2. Study Design

We announced to the students of our university that we will conduct a data collection study for five days using Android smartphones. Our collaborator from the Department of Psychology wanted to have participants who are older than 50 years old as well as below 50 years old in the study. Thus, we gave priority to the applicants who will be able to join the study together with a smartphone user who is older than 50 years old. 22 participants joined the study. 10 of the 22 participants were between 50 and 63 years old. Rest of the them were between 21 and 33 years old. There were 11 university students, two housewives, three retired, two self-employed, one academician, one researcher, one teacher and one education inspector among the participants. Age, occupation and smartphone model of each participant are given in Table 3.1. 10 of the university students joined the study with a relative over 50 years old. In other words each of the 10 participants over 50 years old is a relative of one of the participants below 50 years old. All of the participants signed an informed consent form [Table A.1]. Participants were paid 150 Turkish lira at the end of the five day study.

All of the participants installed the Android application of Smarpthone Tracker system on their smartphones. We then configured all of the smartphones from the dashboard to collect data from 19 different probes. Data collection intervals and durations for these probes are given in Table 3.2. For example, we collected ten seconds of accelerometer data every five minutes and recorded one minute long audio every two hours. We asked the participants to keep GPS and location services of their smartphones enabled all the time.

Our collaborator gave us two surveys for this study namely Thinking and Talking. There are ten questions in the Thinking survey and nine questions in the Talking survey. One minute audio is automatically recorded every two hours in the smartphones of the participants between 10 am and 11 pm everyday. After an audio recording is finished, a survey notification is sent to the user. This mechanism allows us to correlate survey answers with the content of the audio recordings. The first question of a survey was always “Where you talking just before the notification?” [Figure 3.15]. If a user answers this question by tapping the “Yes” button, questions from the Talking survey is asked in the remaining of the survey. If a user taps the “No” button, questions from the Thinking survey is asked. Questions that are used in these surveys are given in Appendix [Table A.2 and Table A.3]. Since all of the participants were speaking Turkish language, we used Turkish translations of these questions in the study.

Results of the study are given in the next section.

Table 3.1. 5 days 22 people data collection study participant list.

ID	**Age**	**Occupation**	**Smartphone Model**	**Android Version**
1	21	Student (BS)	Samsung GT-I8150	4.0.4
2	21	Student (BS)	Samsung GT-I9300	4.3
3	21	Student (BS)	HTC EndeavorU	4.3.1
4	22	Student (BS)	Samsung GT-N7100	4.3
5	23	Student (BS)	Samsung GT-I9500	4.4.2
6	23	Student (BS)	Samsung GT-I9300	4.3
7	23	Student (MS)	HTC Desire HD A9191	2.3.5
8	23	Student (BS)	TURKCELL MaxiPRO5	4.0.3
9	24	Student (BS)	HTC ChaCha A810e	2.3.5
10	26	Student (MS)	Sony C1905	4.1.2
11	30	Researcher	Samsung GT-I9190	4.2.2
12	33	Student (PhD)	Samsung GT-N7100	4.3
13	50	Retired	Samsung GT-S5830i	2.3.6
14	52	Retired	LG-E612	4.0.3
15	52	Housewife	Samsung GT-I8150	2.3.6
16	52	Self-employed	Huawei P6-U06	4.2.2
17	53	Teacher	LG-P970	4.0.4
18	54	Housewife	TURKCELL MaxiPRO5	4.0.3
19	56	Academician	Samsung GT-N7100	4.3
20	57	Retired	Samsung GT-I8190	4.1.2
21	63	Self-employed	Samsung GT-S5360	2.3.6
22	63	Education Inspector	Turkcell Maxi Plus 5	4.0.4

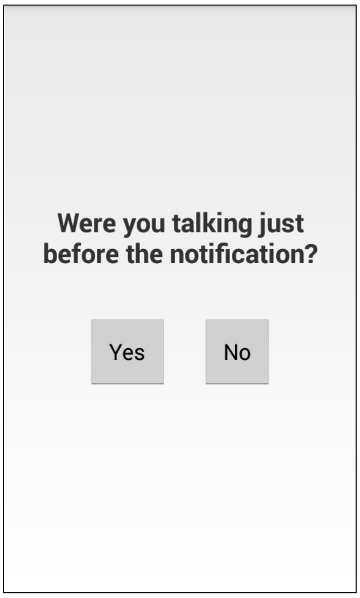


Figure 3.15. Screenshot of the first question of a survey. The first question of the surveys is always the same. The remaining questions are shown either from Talking or Thinking surveys depending on the answer to the first question.

3.2.3. Results

<u>3.2.3.1. Participant Information.</u> Age, occupation and smartphone information of the participants are given in Table 3.1. We obtained age and occupation information from the participants before the study. Model and Android version of the smartphones are automatically collected during the study.

<u>3.2.3.2. Data Completeness.</u> Even tough we informed the participants to keep their smartphones working most of the time, there may have been times when our data collection application may not have been running on the smartphone of a participant. We can estimate how much of the time our application was running on a smartphone from the completeness of the most frequently collected data. With ten seconds of data for every five minutes, accelerometer is the most suitable probe to estimate total running time of our application. We divide five days into 1440 time slots each five minutes long. We assume that our application was working in a five minutes long time slot if there is at least one accelerometer reading in that timespan. Figure 3.16

Table 3.2. Data collected from the smartphones. Interval is the time between two consecutive data requests for a probe. Data are collected at each interval for the given duration. A dash (-) is shown when duration is not applicable for a probe.

ID	**Probe**	**Interval**	**Duration**
1	Accelerometer Sensor	5 m	10 s
2	Android Information	24 h	-
3	Audio Recording	2 h	1 m
4	Battery Information	30 m	-
5	Bluetooth Scan	10 m	30 s
6	Browser Search History	24 h	-
7	Call Log	24 h	-
8	Cell Tower	30 m	-
9	Contacts	24 h	-
10	Hardware Information	24 h	-
11	Installed Applications	24 h	-
12	Light Sensor	10 m	5 s
13	Location	30 m	-
14	Proximity Sensor	10 m	5 s
15	Running Applications	Pooling	-
16	Screen on/off events	Event based	-
17	SMS History	24 h	-
18	Telephony	24 h	-
19	WiFi Scan	10 m	-

shows the running time estimation of Survey application for each of the participants' smartphones for five days.

Survey application may not be running on the smartphone of a participant for three reasons. First, the application may not be installed on a smartphone at all. White regions in the beginning of the study can be explained with this possibility as some of the participants joined the study late. Second possibility is that the application may have a bug which causes it to crash. However, none of the participants reported such a case. Third possibility is that the smartphone itself may not be working due to an intentional shut down by the user or an empty battery. Some of the white regions in the middle of the study correspond to times when the battery of a smartphone is empty. For example, in Figure 3.17 we see that first participant's smartphone's battery is empty just before the beginning of Day 4. This fact corresponds to a white region in the first participant's data completeness row at that timespan [Figure 3.16]. We can also observe that 19th participant turns his/her phone off during nights for two days.

Normally in Figure 3.16 we either expect continuous black or continuous white regions for hours. However, 2nd, 6th, 8th and 16th participants' rows have regions that frequently alter color between black and white in the figure. 2nd and 6th participants have the same smartphone model namely Samsung GT-I9300. These two participant also reported various problems during the study. We suspect that these problems are related their smartphone model instead of the Android version since there are other smartphone models that runs Android version 4.3 without a problem in the study. If we were able to collect data all the time in five days, we would expect 120 hours of data from each of the smartphones. Right axis of Figure 3.16 shows the total running time of our application for each of the smartphones in hours. From the 22nd participant we were able to collect only 20 hours of data. 17th participant uninstalled our application in the middle of the study. Despite all of the white regions that correspond to times that we were unable to collect data from the smartphones, we were able to collect on average 82 hours of data per person in five days. We were able to capture daily smartphone usage patterns in various data modalities. These data modalities will be investigated in the following sections.

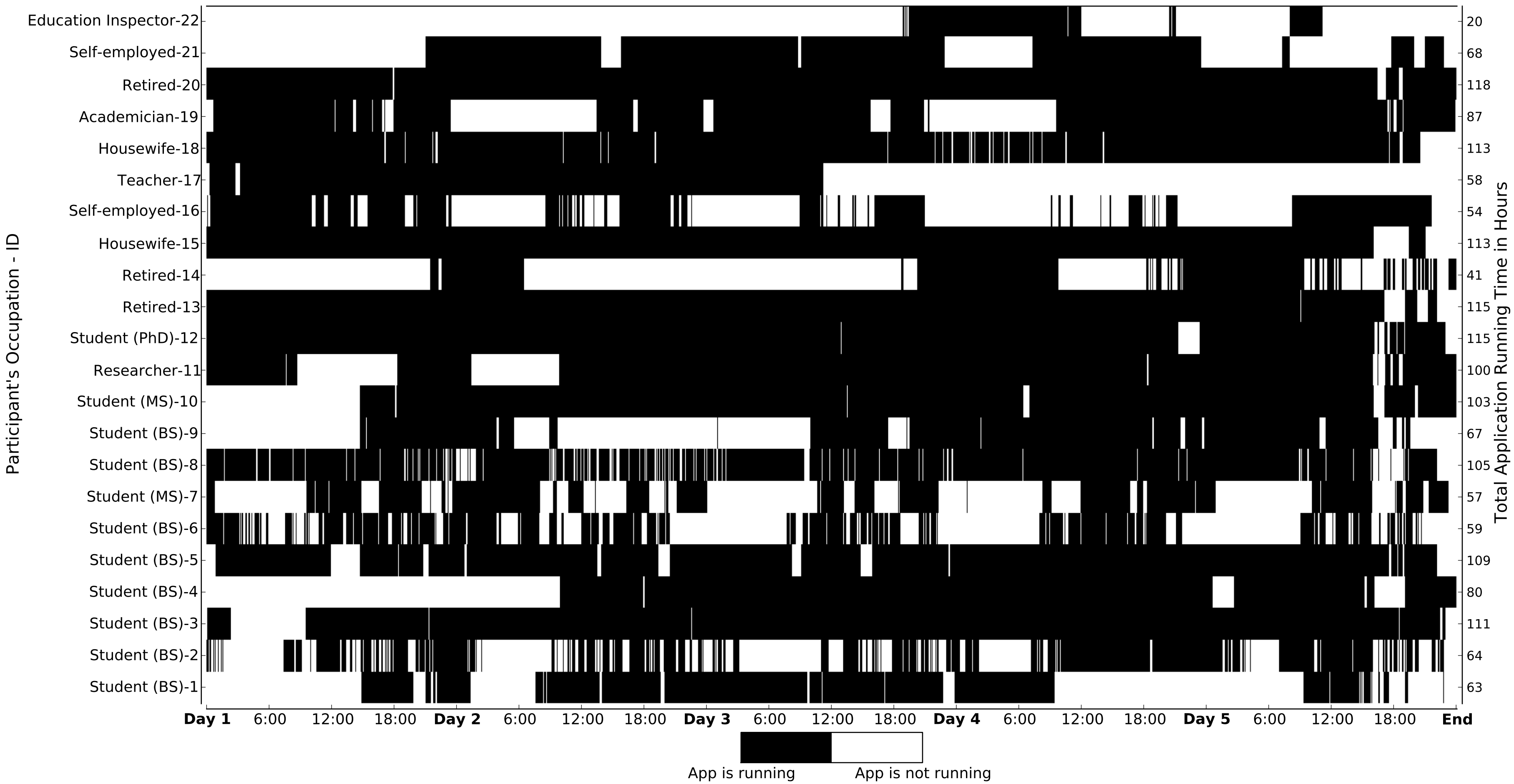


Figure 3.16. Survey data collection application running time estimation. We assume that the application is running in a five minutes timespan, if there is an accelerometer reading in that timespan.

3.2.3.3. Battery. Battery levels of smartphones are recorded with 30 minutes intervals. Android OS gives the battery level of a smartphone as a value between 0 and 100. 0 means the battery of the smartphone is empty. 100 means the battery of the smartphone is full. In Figure 3.17, each row shows the battery levels of a smartphone in the study for five days. The line on top of a row corresponds to full battery level and the line on the bottom corresponds to empty battery level. For example, first participants battery is empty just before the beginning of Day 4.

We informed participants before the study to charge their smartphones regularly in order to prevent shutdowns due to empty battery. Average battery level of all the smartphones for five days is 62. Average battery level of 15th participant's smartphone is 77. She never let her smartphone's battery to drop below 25. We can observe the most regular battery level pattern in 12th participant's smartphone. His/her smartphone's battery lasted almost one day after every charge during the five days. After the study some of the participants reported that their smartphone batteries endured less during the study compared to times before the study. We expect this as we collected various types of data from the smartphones periodically. However, 4th, 10th, and 12th participants mentioned they haven't experienced any serious battery problems with their relatively new smartphone models.

3.2.3.4. Android Information. Our data collection application supports devices having Android version 2.3 or above. Android versions of the smartphones in the study are shown in Table 3.1. The oldest Android version among the smartphones is 2.3 and the newest version is 4.4. Android Compatibility Definition Document (CDD) contains the software and hardware requirements that an Android compatible device should satisfy [12]. These requirements are updated with the release of a new Android version. For example, devices compatible with Android 4.0 should be able to deliver accelerometer data at 50 Hz or greater, whereas 120 Hz or greater is recommended for devices that complies with Android 4.4. Android version information is useful for debugging purposes when a problem is experienced with any of the smartphones during a data collection study.

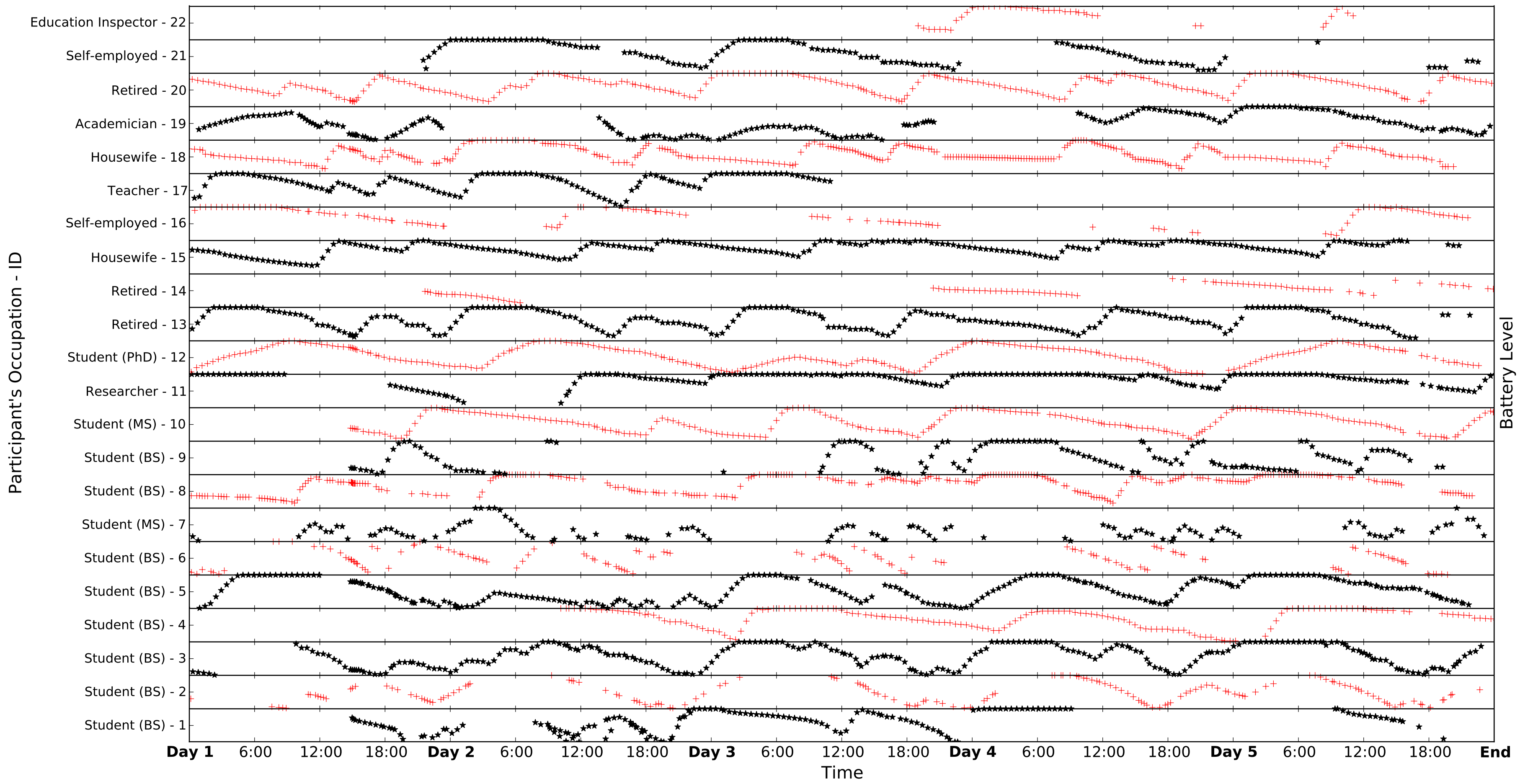


Figure 3.17. Battery levels of the participants' smartphones for five days.

3.2.3.5. Hardware Information. Hardware information of a smartphone contains brand, model, Bluetooth MAC address, and Wi-Fi MAC address of the smartphone. Bluetooth MAC address is useful to discover proximity between participants of a data collection study. For example, we discovered that 15th participant's smartphone found 6th participant's smartphone's Bluetooth MAC address in 140 Bluetooth scans conducted during the study. We know that these participants are relatives who joined the study together. Brand and model of the smartphones are useful to determine the participants who have the same smartphone model. In this study, we have two Samsung GT-8150, two Samsung GT-I9300, three Samsung GT-N7100 and two TURKCELL MaxiPRO5 smartphones.

3.2.3.6. Accelerometer. We requested accelerometer data during ten seconds for every five minutes from each of the smartphones in five days. Figure 3.16 also shows the times when accelerometer data are available from each of the smartphones since we use the completeness of the accelerometer data to estimate the application running times.

Android OS allows applications to acquire sensor data from the built-in accelerometer of a smartphone by using the Android Sensor API. There are four different sensor delay types to choose from in the Sensor API. These delay types and corresponding delays are given in Table 3.3. Chosen delay type determines the rate at which the accelerometer values will be sampled. The delay type is only a suggestion to the Android OS [53]. Different smartphone models may have different sampling rates due to different hardware and software on them.

Table 3.3. Android Sensor API delay types [53].

Sensor Data Delay Type	**Delay in Microseconds**
SENSOR_DELAY_NORMAL	200,000
SENSOR_DELAY_UI	60,000
SENSOR_DELAY_GAME	20,000
SENSOR_DELAY_FASTEST	0

SENSOR_DELAY_FASTEST delay type is used in our data collection application. With this delay type we request accelerometer data at the maximum possible sampling rate from each of the smartphones. Figure 3.18 shows the average accelerometer sampling rates of 22 smartphones for five days. We observe that six of the smartphones have an average accelerometer sampling rate below 50 Hz.

There are three Samsung GT-N700, two TURKCELL MaxiPRO5 and two Samsung GT-I9300 smartphones in the study. Figure 3.19 shows sampling rate histograms for these smartphones. We would expect to see 1440 data requests in total in each of the histograms if we could collect data from the smartphones for every five minutes in five days. However, due to the reasons mentioned in Section 3.2.3.2, our data collection application could not collect accelerometer data all the time. We can see that smartphones with the same model have similar sampling rate histograms. We can also observe the problematic behavior of the smartphone model of the second and sixth participants namely Samsung GT-I9300. Normally we expect to see a peak at a certain sampling rate in each of the histograms which would approximately correspond to the maximum accelerometer sampling rate of the smartphone. For example, from three Samsung GT-N7100 smartphones we get 100 Hz of accelerometer data as a result of most of the data requests. Similary from two TURKCELL MaxiPRO5 smartphones we get approximately 50 Hz of data most of the time. However, from two Samsung GT-I9300 smartphones we get data with a wide range of sampling rates without a peak.

Even tough the smartphone model of the 8th and 18th participants namely TURKCELL MaxiPRO5 samples accelerometer data with 50 Hz, it complies with the requirements of Android. We can see from Table 3.1 that the Android version of these smartphones is 4.0.3. Android Compatibility Definition Document for version 4.0.3 specifies that compatible devices should be able to deliver accelerometer data at 50 Hz or greater [12].

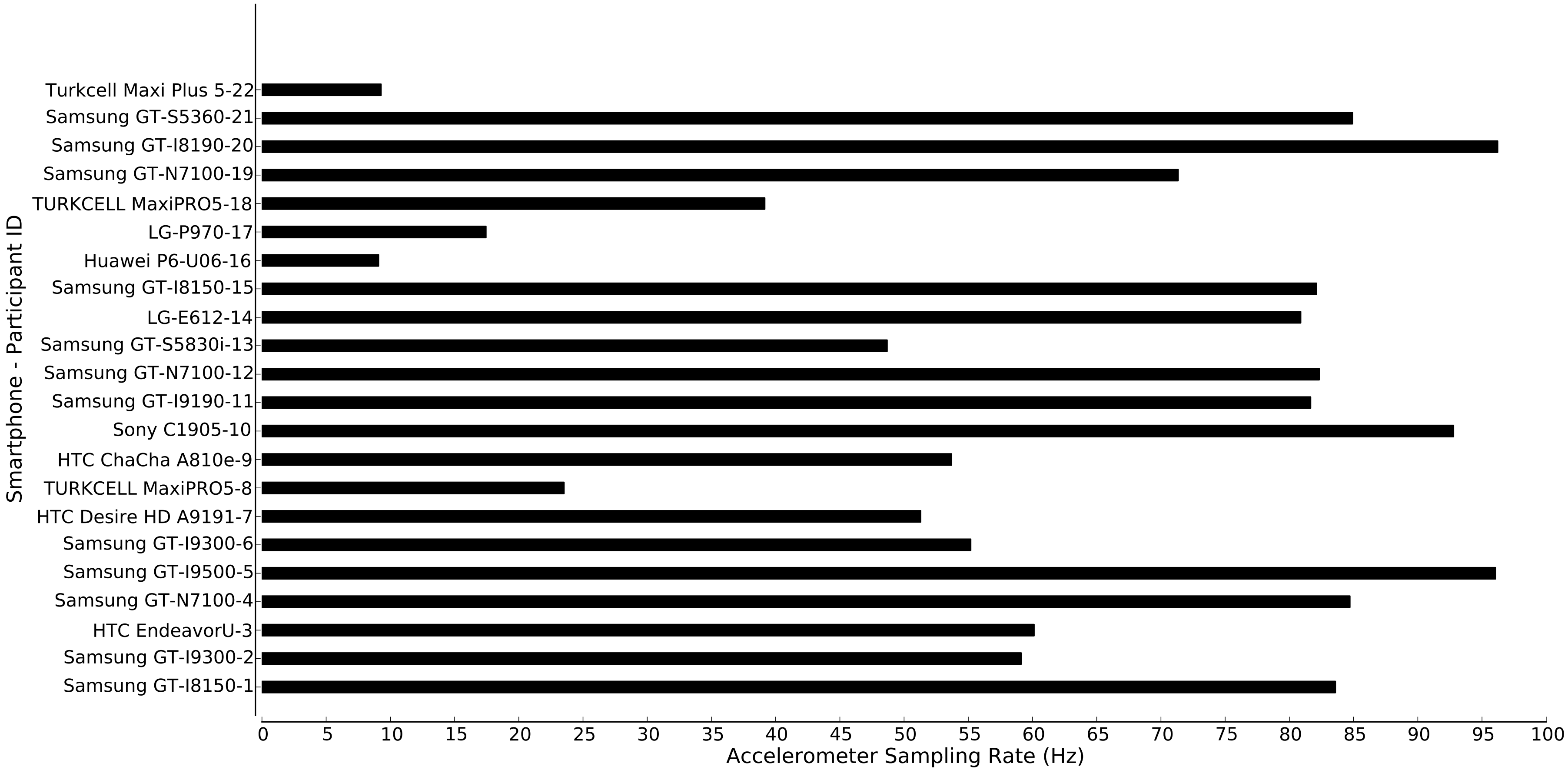


Figure 3.18. Average accelerometer sampling rates of the smartphones for five days.

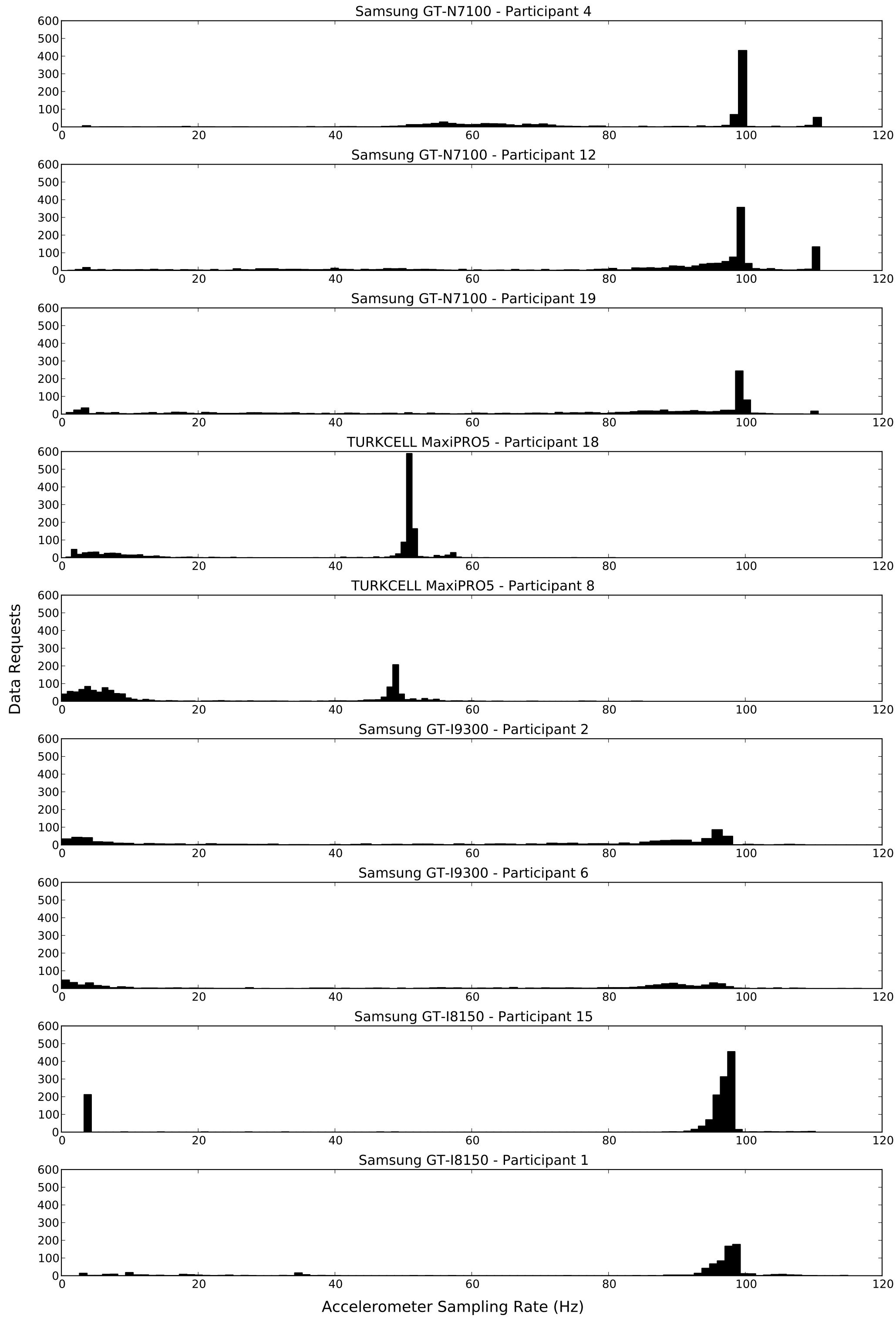


Figure 3.19. Accelerometer sampling rate histograms for Samsung GT-N7100, TURKCELL MaxiPRO5, Samsung GT-I9300 and Samsung GT-I8150 smartphones in the study.

Accelerometer data can be used to infer the activity level of a smartphone user. Variance of the accelerometer magnitude values can be used as an indicator of the physical activity level [39]. Figure 3.20 shows the activity levels of a participant during the study. Magnitude of one accelerometer reading is computed as in Equation 3.1. The variance of the magnitude which corresponds to the activity level in each duty cycle is calculated as in Equation 3.3 where m_i is the ith magnitude, N is the number of accelerometer readings in that duty cycle and $\bar{m}$ is the mean of the magnitudes calculated as in Equation 3.2. a_x, a_y and a_z are the acceleration values on x, y and z axes respectively. In Funf framework's activity probe, if the variance is above 10, activity level is determined as high; if the variance is between 10 and 3, the level is determined as low; if the variance is below 3, the level is determined as none. These thresholds can roughly be observed in our raw accelerometer data analysis in Figure 3.20.

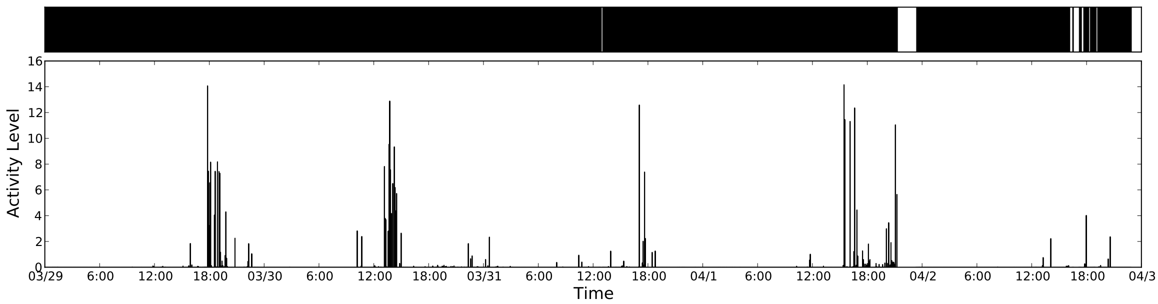


Figure 3.20. Physical activity levels of a participant during the study. The bar on top shows the times when the accelerometer data are available as black regions. In the white regions no accelerometer data is available.

$$m = \sqrt{a_x^2 + a_y^2 + a_z^2} \tag{3.1}$$

$$\bar{m} = \frac{1}{N}\sum_{i=1}^{N} m_i \tag{3.2}$$

$$s_N^2 = \frac{1}{N}\sum_{i=1}^{N}(m_i - \bar{m})^2 \tag{3.3}$$

3.2.3.7. Wi-Fi. A Wi-Fi scan is conducted every ten minutes by each of the smartphones. Figure 3.21 shows the times when there is at least one Wi-Fi access point nearby the smartphones. Right axis of the Figure 3.21 shows the nearby AP presence time over application running time ratio for each of the participants in the study. We can observe the high availability of Wi-Fi infrastructure for most of the participants. We know that, as it will be explained in the multimodal analysis of the study in Section 3.2.3.14, 15th and 18th participants who are housewives were at home most of the time during the study. We observe in Figure 3.21 that at least one AP is found nearby of these participants most of the time during the study. For 3rd, 9th and 13th participants we observe that most of the time the application is running but no AP is found nearby. 3rd participant reported that s/he used Wi-Fi during the study. However, our data collection application did not log any Wi-Fi scans for this participant. Thus, there was a problem with Wi-Fi activity collection from the 3rd participant's smartphone. We were able to collect other types of data from the smartphone of this participant such as accelerometer and call logs without experiencing a problem.

3.2.3.8. Bluetooth. A bluetooth scan is performed every ten minutes for 30 seconds by each of the smartphones in the study. Figure 3.22 shows the times when there is at least one bluetooth device nearby the smartphones. The number of devices found in a bluetooth scan can be used as an indicator of the number of people in the place that the smartphone is located at. Right axis of the Figure 3.22 shows the average number of bluetooth devices found in a scan for each of the smartphones in the study. For the 19th participant's smartphone, even tough there is at least one bluetooth device nearby most of the time, average number of bluetooth devices found in a scan is 1.6 which is relatively low compared to other participants' smartphones. The reason is that a bluetooth device named "TVBluetooth" was found in most of the scans. This device is probably a television with a bluetooth technology that the 19th participant owns. Similarly for the 15th participant, a bluetooth device with a name "DTVBluetooth" and her relative's smartphone which is also one of the smartphones in the study were found most of the time. On average 10.8 bluetooth devices were found by the smartphone of the 11th participant which is the highest number in the study.

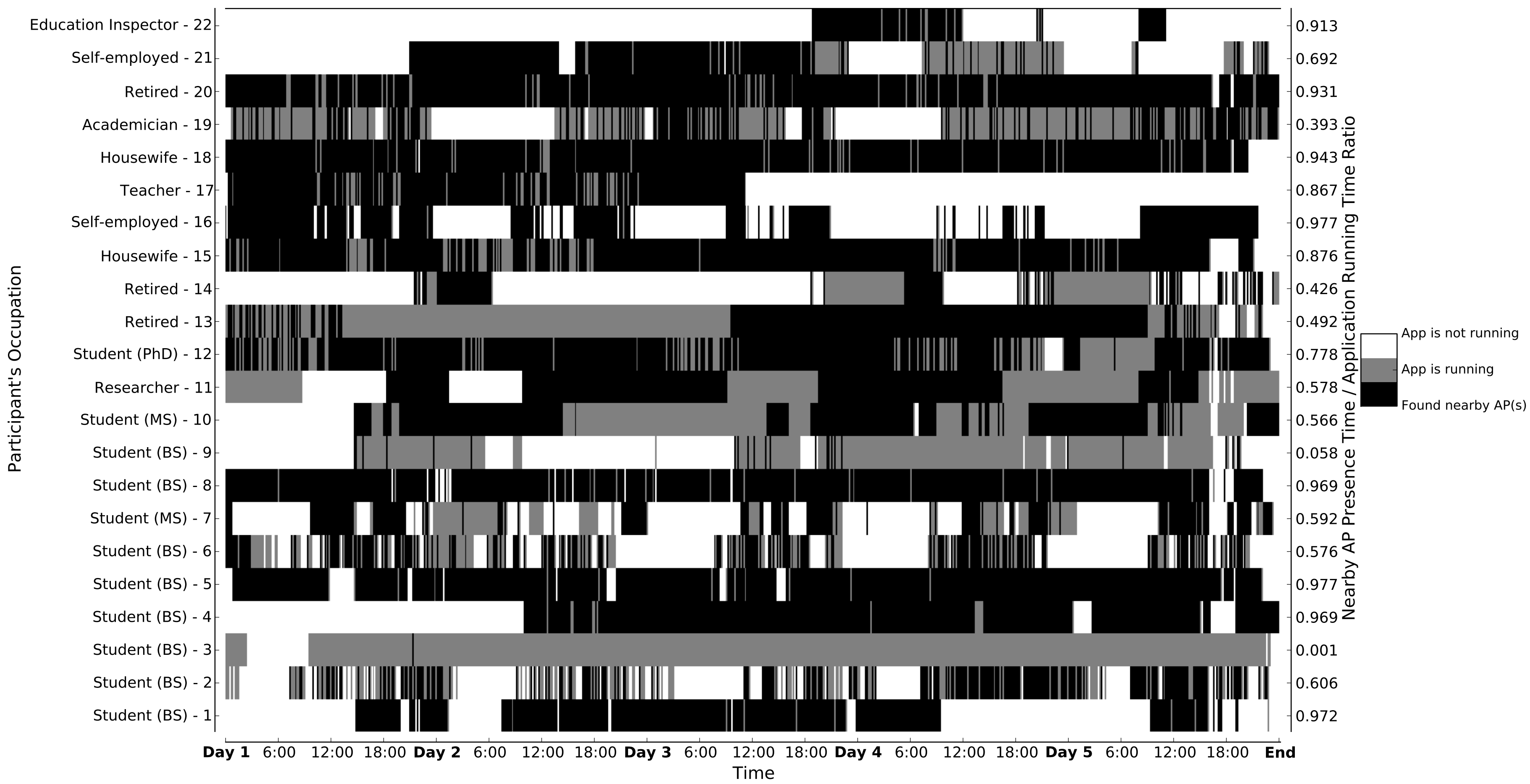


Figure 3.21. Wi-Fi scan results. The ratio of nearby AP presence time to application running time is shown on the right axis.

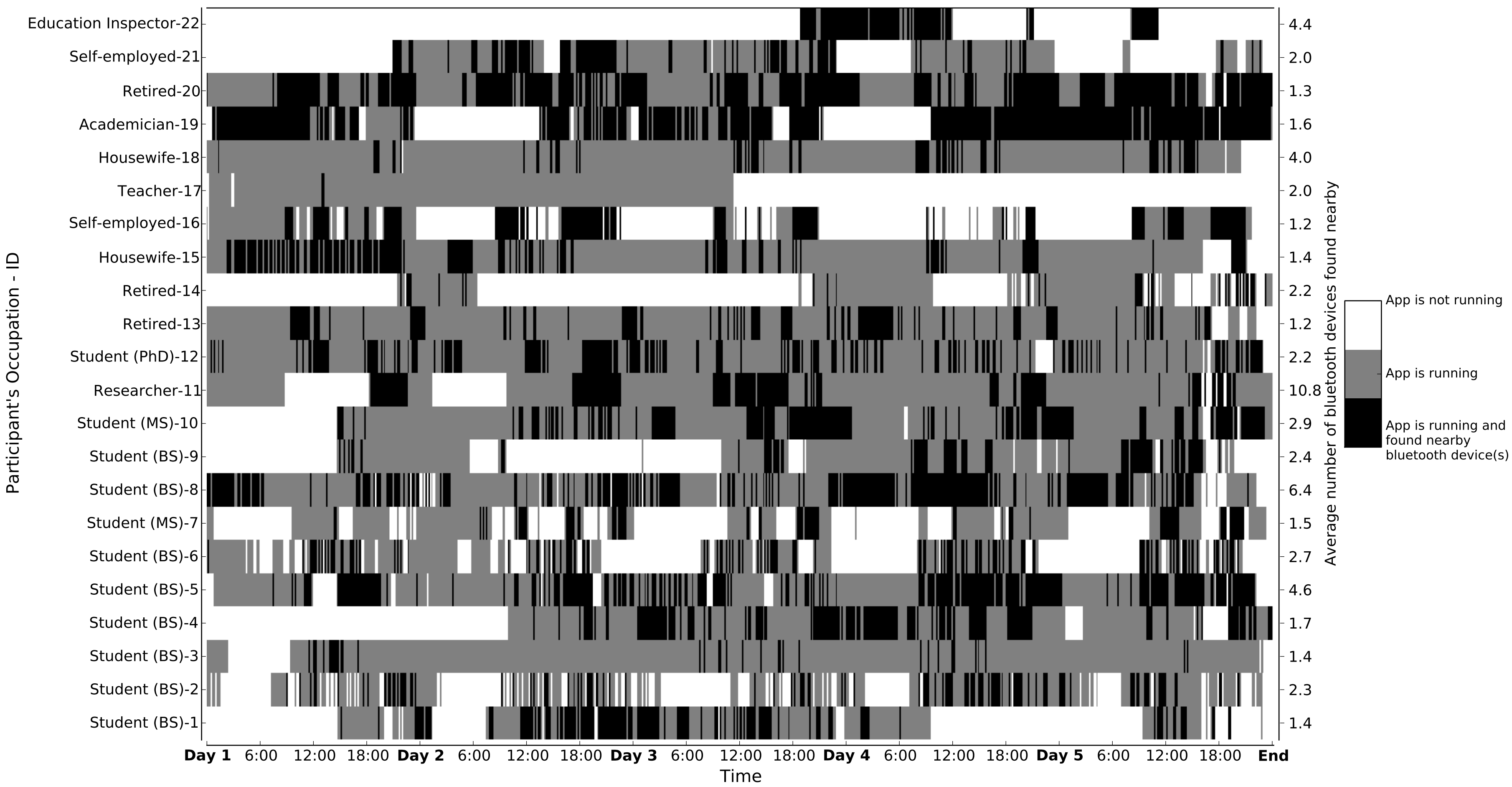


Figure 3.22. Bluetooth scan results. Average number of blutooth devices found in a scan is shown for each device on the right axis.

3.2.3.9. Call Log. Call log of the participants are collected daily. Type (i.e., incoming, outgoing and missed), time, and duration of the calls and one way hashed values of the phone numbers used in the calls are collected from the smartphones. Figure 3.23 shows the call log of the participants during the study. We can observe that the participants made phone calls mostly between 6:00 and 24:00 on each day during the study. Right axis of the Figure 3.23 shows the total number of call log events for each of the participants. These events include incoming, outgoing and missed calls. We can see that 181 call log events happened on the smartphone of the 22nd participant which is the highest number in the study. Android OS gives the complete call log history of a smartphone if the user did not explicitly delete it. Thus, we are able to observe the call log pattern of the 22nd participant even tough our data collection application ran only 20 hours on the smartphone of this participant in five days.

3.2.3.10. SMS Messaging History. SMS messaging history of the participants are collected daily. Type (i.e., incoming, outgoing), time, and one-way-hashed values of the phone numbers used in the SMS messages are collected from the smartphones. Figure 3.24 shows the SMS messaging history of the participants during the study. Right axis of the figure shows the total number of SMS messages sent and received for each of the participants. We observe less SMS events for participants over 50 years old compared to the participants below 50 years old. The participants over 50 years old sent 35 messages and received 177 messages in total. The participants below 50 years old sent 420 messages and received 596 messages in total. We also see less SMS messaging events compared to the call log events for the participants over 50 years old. We observe that for the 15th participant there is no messaging event. This participant may be clearing her SMS messaging history manually. 1st and 9th participants used SMS significantly more than telephony whereas 3rd participant used telephony more. 22nd participant who has the most phone call events during the study, never send an SMS message even tough s/he received 19 messages. We can observe the high SMS and telephony usage of the 19th participant compared to the other participants. We also observe more SMS messaging events compared to call log events between 00:00 and 06:00 during the study.

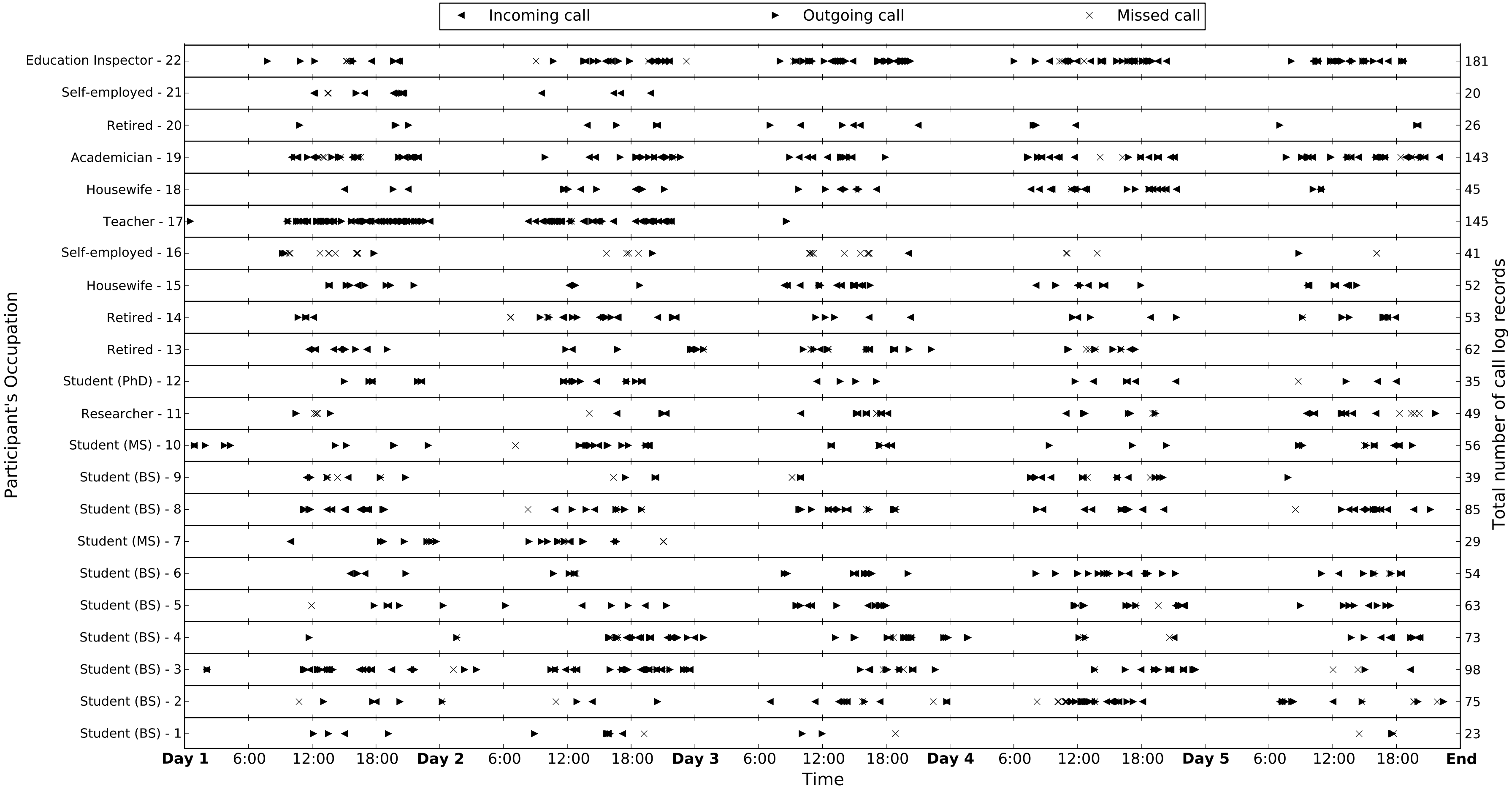


Figure 3.23. Call logs of the participants. Total number of call log events are shown for each participant on the right axis. The phone calls are made mostly between 6:00 and 24:00 on each day during the study.

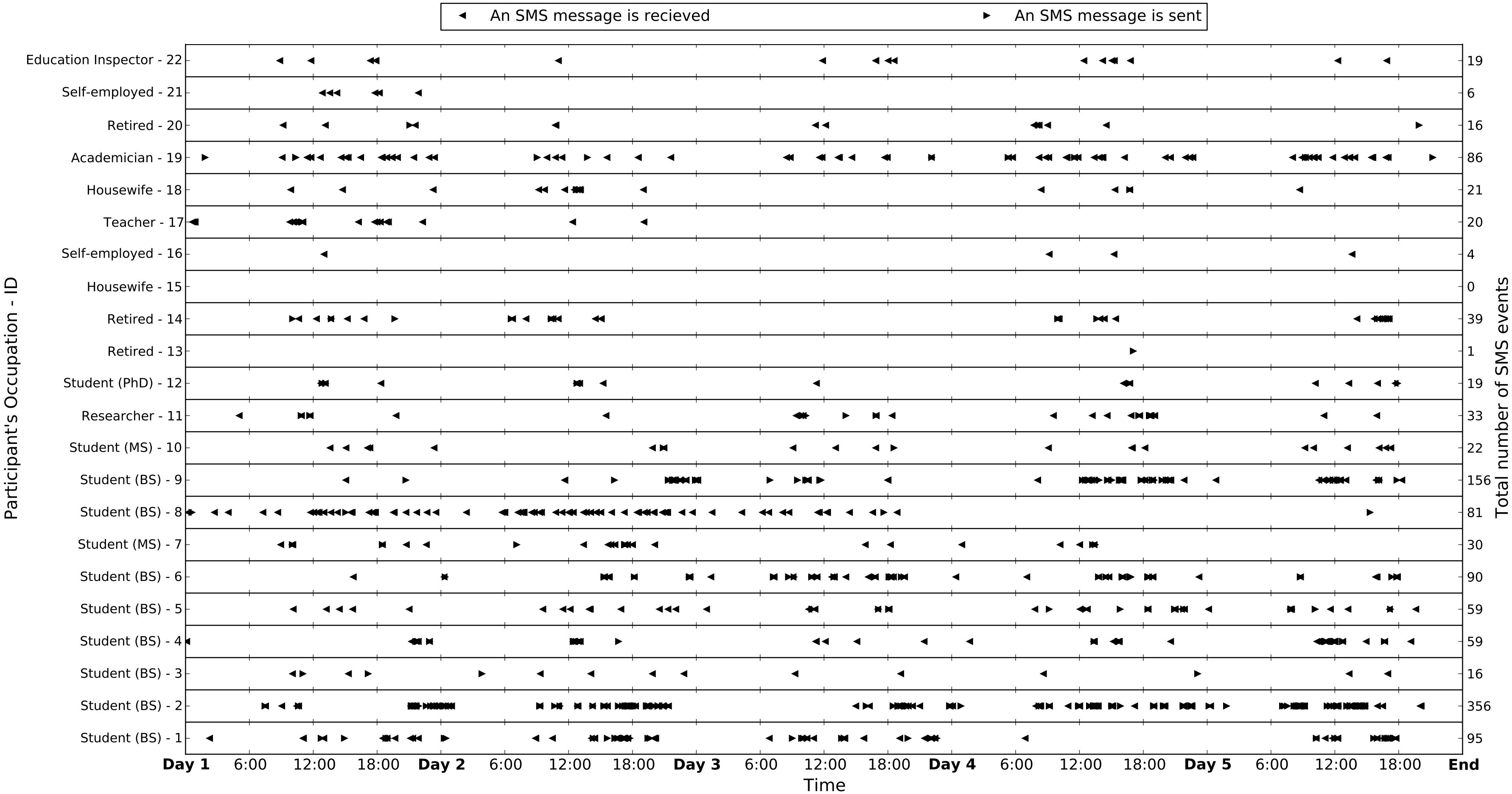


Figure 3.24. SMS messaging history of the participants. Total number of SMS events are shown for each participant on the right axis.

3.2.3.11. Audio Recording. One minute audio is recorded automatically every two hours from the smartphones during the study. The participants were able to listen the recorded audio files and choose which ones to upload to server using the interfaces shown in Figure 3.6. Figure 3.25 shows the finish times of audio recordings. There was a bug in our implementation of audio recording scheduling mechanism. Even tough we configured smartphones to record audio every two hours, audio recordings were performed more frequently. We recorded audio only between 10 am and 11 pm. However, we observe five audio recordings before 10 am on the row of the 11th participant on the first day of the study. Because the 11th participant's smartphone had an earlier version of our application in the beginning of the study which was allowed to record audio at any time during 24 hours. After the first day of the study s/he also installed the latest version of our application.

After an audio recording is finished, a survey notification is sent to the user. Our purpose with this mechanism is to correlate survey answers with the content of the audio recordings. Our collaborator from the Department of Psychology will analyze the survey answers and listen the audio recordings to find correlations between them. One of our findings is that the user of a smartphone may not respond to a notification immediately. S/he may even delete the notification from the notification area of the smartphone without responding it. Thus any mechanism that requires user intervention should consider the notification time and the time when the user responded the notification. One possible solution to make use of most of the user responses is to remove a notification after a timeout period. For example if a user does not respond to a notification in one minute, notification can be removed from the notification area of the smartphone automatically. We make use of this type of prevention in our Auto Diary application which is introduced in Chapter 5. In this application, we discover locations important for the user of a smartphone using the Wi-Fi infrastructure. When a new location is discovered, a notification is sent asking the user of the smartphone to give a name to his/her current place. However, if the user leaves the location and later responds to the notification in another location, his/her response would not be valuable as the notification does not belong to his/her new location. Thus the notification is removed from the notification area of the smartphone if the user leaves his/her current location before responding to the notification.

3.2.3.12. Survey. In Figure 3.25, survey submission times of participants are shown. After a one minute audio recording is finished, a survey notification is sent to a user immediately. Since we record audio only between 10 am and 11 pm, the participants received survey notifications only at this time interval. Duration between a survey notification and a survey submission is considered as delay. For each survey submission most recent audio recording finish time is considered as the notification time. On the right axis of Figure 3.25 mean delay in survey submission is shown for each participant. The delay includes the time required to fill the surveys. We observe that both of the housewives namely 18th and 15th participants responded to most of the survey notifications and have a delay under ten minutes. We observe a delay of 44 minutes in 21st participant's submissions. S/he also submitted only seven surveys during the study. The questions used in the surveys are given in Appendix [Table A.2 and Table A.3]. The submitted survey answers will be analyzed by our collaborator from the Department of Psychology at University of Zurich. We only use the answers of the participants to the "Where are you" question as a ground truth of the partcipants' locations in the multimodal analysis of the collected data in Section 3.2.3.14. As an answer to this question participants were able to choose one of the six predefined choices namely Home, Work, In a vehicle, Restaurant/Cafe/Bar, Supermarket and Outdoor/Public Place. As a future work, we will improve the survey system to enable the participants enter their custom location labels as an answer to this question.

3.2.3.13. Other Probes. The location of a smartphone can be inferred from the location of the connected cell tower [34]. There are publicly available crowd-sourced databases that map cell tower IDs to physical locations. Cell tower probe collects the connected cell tower information which includes the cell tower IDs. In addition to cell tower ID, network operator code and country code are required to query these databases. Telephony probe collects this type of information such as mobile country code, mobile network operator code and one-way-hashed value of phone number. One-way-hashed value of the phone number can be used to determine phone calls be-

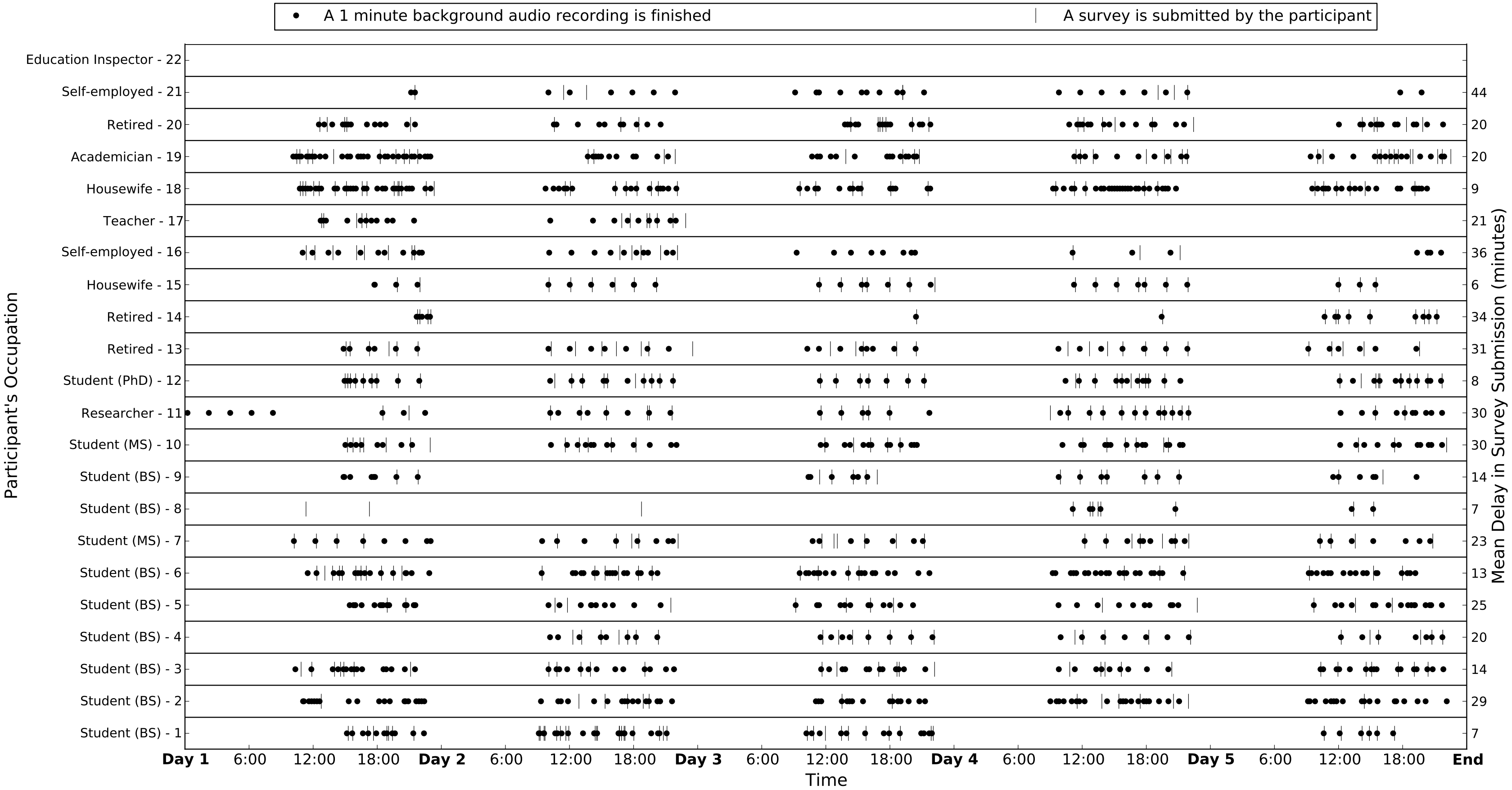


Figure 3.25. Delay in the survey submissions. After a one minute audio recording a survey notification is sent immediately. Duration between a survey notification and a survey submission is considered as delay. For each survey submission most recent audio recording finish time is considered as the notification time. Mean delay in survey submission is shown for each participant on the right axis.

tween the participants of a smartphone sensing study. One-way-hashed values of phone numbers used in phone calls can be compared with each participants' one-way-hashed phone number to determine the parties involved in a phone call. The collected information from Contacts probe includes the hashed values of the phone numbers, notes and names used for each contact. Information obtained from Contacts probe can be used to discover social networks among the participants in a smartphone sensing study by looking at shared contacts of the participants. Browser probe collects the information of when a search was performed and the one-way hashed values of the terms used in the search. Only six of the participants used browser search at least one time during our study. However, Android OS gives only the search history of its native browser and the participants may be using third-party browsers. Running applications and installed applications probes collect the information about the applications used by a smartphone user. Light and proximity sensor probes together can be used to determine the lightning of the environment the user is in. We can detect when there is an object in front of the smartphone which blocks its light sensor's view by using the proximity sensor and discard values obtained from the light sensor at that times. Screen on/off probe can be used to determine the smartphone interaction times of the participants during a day.

3.2.3.14. Multimodal Analysis of the Study. In Figure 3.26, battery level, data availability, physical activity level, Wi-Fi activity and survey answers from the 12th participant's data are shown together. In order to be able to visualize the data clearly, we use the timespan between Day 3 and Day 5 for the analysis.

Answers of the participant to the "Where are you?" survey question are shown on the sub-figure with a "Survey Location" label on its y axis. As an answer to this question, a participant was able to choose one of the six predefined location values seen in the y axis of the figure. For example, the participant was at supermarket around 18:00 in Day 3.

The y axis of the BSSID ID figure assigns a unique ID to each of the BSSIDs

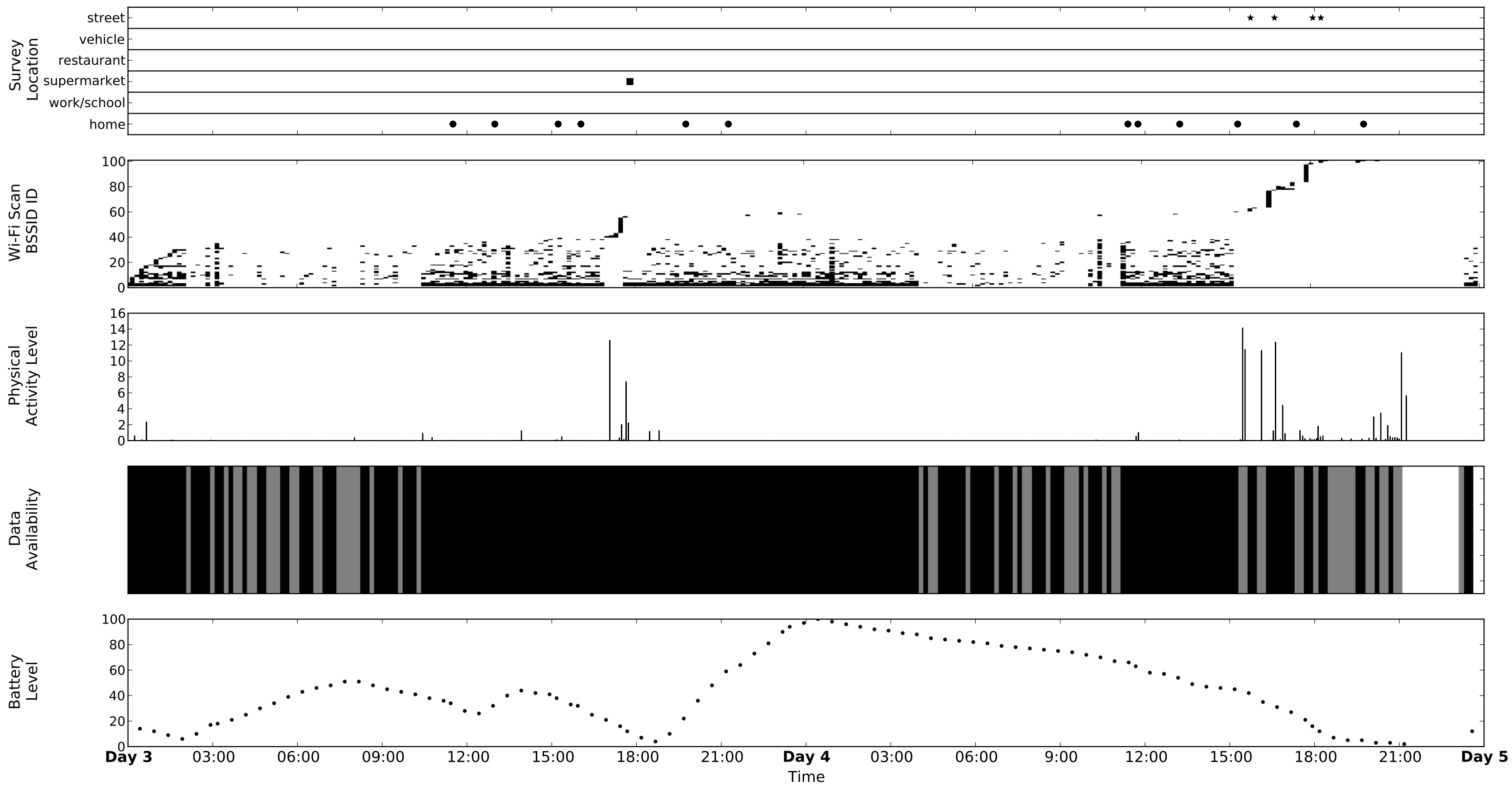


Figure 3.26. Multimodal analysis of the 12th participant's data. For example, the user reported his/her location via a survey that s/he is at supermarket around 18:00 in Day 3. We also observe high physical activity around that time and new BSSIDs are introduced as the user enters new places. The battery of the smartphone is empty for about three hours after 21:00 in Day 4.

found in the Wi-Fi scans between Day 3 and Day 5. The black regions in this figure shows the times when a BSSID is found in a Wi-Fi scan. For example, we see black regions most of the time at the bottom of the figure which probably correspond to times when the user is at home and the APs nearby of the user's home are found during Wi-Fi scans. We can verify this assumption using the surveys submitted by the participant. The user submitted two surveys around 12:00 in Day 3 in both of which s/he mentioned that s/he is at home. In the BSSID ID figure we observe that BSSIDs with an ID between 0 and 40 are found in the Wi-Fi scans around the time when the user submitted the surveys. Thus, we infer that these BSSIDs belong to APs that are located nearby the user's home.

The activity level figure is generated from the variance of the acceleration magnitude values of the smartphone as it is described in Section 3.2.3.6. We observe high activity around 18:00 in Day 3, between 15:00 - 18:00 in Day 4, and around 21:00 in Day 4. We observe no activity most of the time, for example, the hours around 06:00 in both of the two days.

The battery figure shows the battery levels of the smartphone for two days. We observe that the battery of the smartphone is full at the beginning of Day 4. It decreases gradually after the beginning of Day 4 and finally it is empty for about three hours after 21:00. When the battery of a smartphone is empty, the smartphone does not work and naturally our application cannot perform data collection.

The data availability figure corresponds to the 12th participant's row in Figure 3.21 where we show the Wi-Fi availability for all of the participants. The gray regions show the times when the data collection application is running. Black regions show when the application is running and at the same time when nearby APs are found during periodic Wi-Fi scans. We observe that most the time there are APs nearby of the participant in two days. We also observe that the application is not running and naturally no APs are found when the battery of the smartphone is empty for about three hours after 21:00 since it could not conduct Wi-Fi scans.

There are fluctuations in the Wi-Fi availability between 01:00 - 11:00 in Day 3, 04:00 - 11:00 in Day 4 and 15:00 - 21:00 in Day 4. The first two timespans (i.e., 01:00 - 11:00 and 04:00 - 11:00) mostly correspond to night time. Android OS looks for idle times and disables some of its services to reduce energy consumption. We suspect that during night time this participant's smartphone enters into sleep mode in which it disables some of its services. Even tough our data collection application schedules periodic system alarms to wake-up the CPU in order to perform its operations such as conducting a Wi-Fi scan, Android OS does not guarantee that the alarms will be delivered at exact times. For example, an alarm scheduled at the top of every hour may be delayed for almost a full interval [54]. During the time between 15:00 - 21:00 in Day 4, we observe high activity in the activity level figure. In addition, the user submitted four surveys in which s/he mentioned "street" as his/her location during this timespan. Thus, the fluctuations between 15:00 - 21:00 may be due to the mobility of the user and unavailability of Wi-Fi infrastructure around him/her on the street.

Finally, we observe a high correlation between location, BSSID ID and activity level figures in terms of the mobility of the user. When there is a high activity, we also observe a change in the BSSID ID figure. For example, around 18:00 in Day 3, activity level of the participant is high. In addition, BSSIDs with an ID between 0 and 40 are no longer found in the Wi-Fi scans. That means the user left his/her home. We also have the ground truth obtained from the surveys that the user went to supermarket around 18:00. Thus, we are able to capture the mobility of the user in three different data modalities. We observe the same correlation between these modalities between 15:00 - 21:00 in Day 4. The user reported that s/he is on the street in four surveys during that time. We also observe high activity and new BSSIDs are introduced as the smartphone enters new areas together with the user.

In Figure 3.27, multimodal analysis of the fourth participant's data is given. The user reported his/her location via surveys that s/he is at work/school between 13:00 and 18:00 in Day 3 and between 11:00 and 14:00 in Day 4. We observe the high correlation between the physical activity and the BSSID ID figures. For example, around 9:00 in Day 4 there is high physical activity and we see an abrupt change in the BSSID figure.

In Figure 3.28, multimodal analysis of the 15th participant's data is given. One interesting event in this participant's data is that the user mentioned his location as street around 15:00 in Day 4. However, we do not observe any change in BSSID ID and physical activity sub-figures. One possibility for this situation is that the user may have left home without taking his/her smartphone with him/her. Another possibility is that the user mistakenly chose the street as an answer to the "Where are you question?". We may considering improving the survey system and the content of the questions to clarify this type of situations. For example, we may add a new question that asks whether a user was carrying the smartphone with him/her.

In Figure 3.29, multimodal analysis of the 18th participant's data is given. Around 14:00 in Day 3 the user submitted two surveys. First, she mentioned that she is at supermarket. In the second survey, she mentioned she is at home. We observe the high physical activity and the change in the Wi-Fi data between these two survey submissions. Around 18:00 in Day 3, we observe high physical activity and the user mentioned her location as street. However, similar BSSID IDs are found around the survey submission time. Even tough the user is highly active, she is stationary with respect to Wi-Fi access points around her during this timespan. For example, around 12:00 in Day 4 we see high physical activity shortly and we also observe a change in the existence of BSSID IDs nearby the user.

As a conclusion, we are able to capture the mobility of the users from accelerometer and Wi-Fi data. Having the ground truths from survey submissions about the users' locations allows us to validate place changes of the users. In addition, we are able to associate BSSID IDs with users' locations such as home and office. However, since a user can only select predefined values as an answer to the "Where are you?" question in surveys, we are limited in accurately determining the user's location from the Wi-Fi data. For example, a user has to submit his/her location as supermarket, even if she goes to different supermarkets. In Chapter 4, we propose a novel localization algorithm which detects stable Wi-Fi environments around the user during a day and requests a place name from the user in real time. We show that the algorithm is able to detect place entrance and departure events accurately from Wi-Fi data.

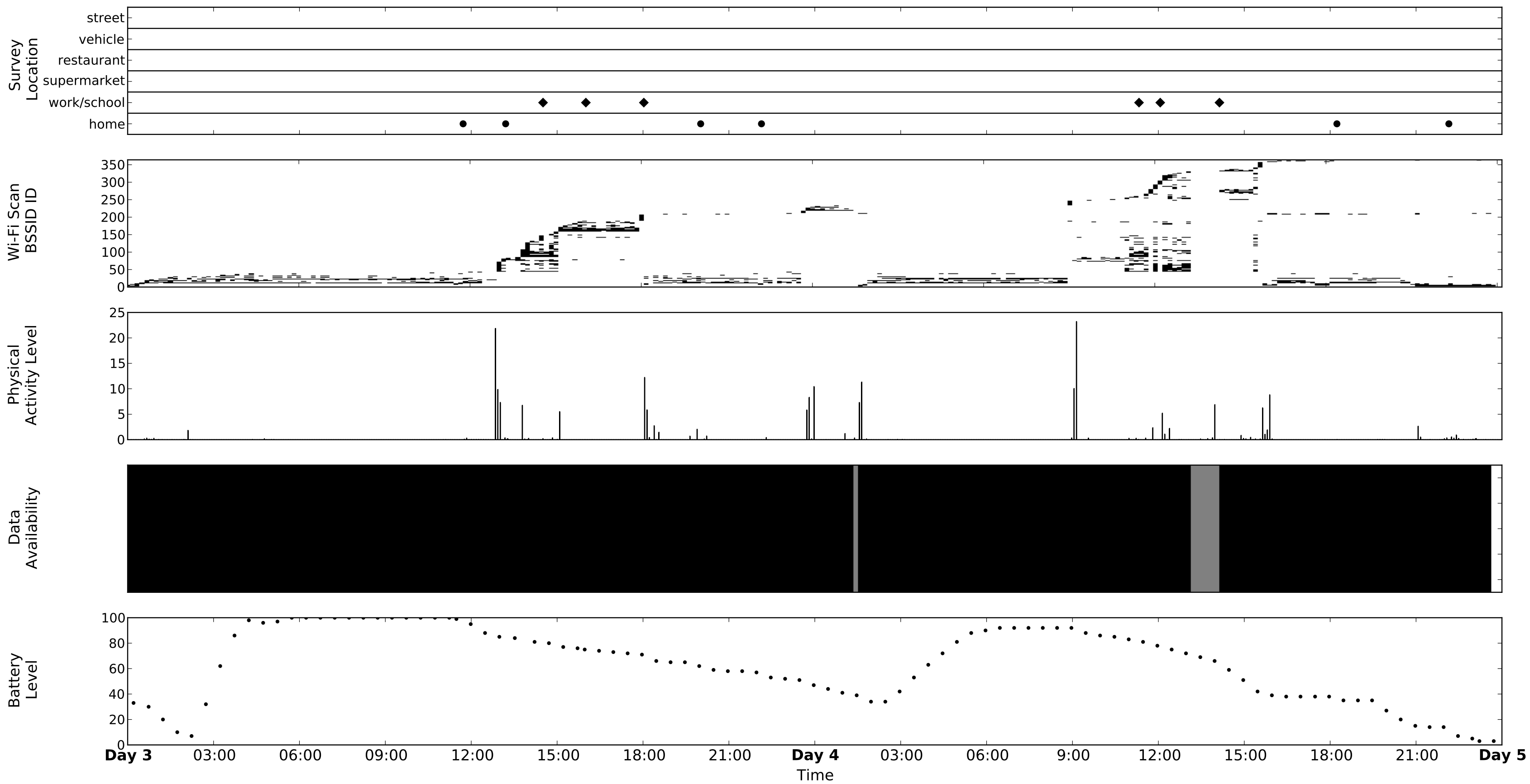


Figure 3.27. Multimodal analysis of the fourth participant's data. For example, the user reported his/her location via surveys that s/he is at work/school between 13:00 and 18:00 in Day 3 and between 11:00 and 14:00 in Day 4. Mobility of the user is captured in surveys, physical activity and Wi-Fi data. For example, the user left home around 9:00 in Day 4 and we observe high physical activity.

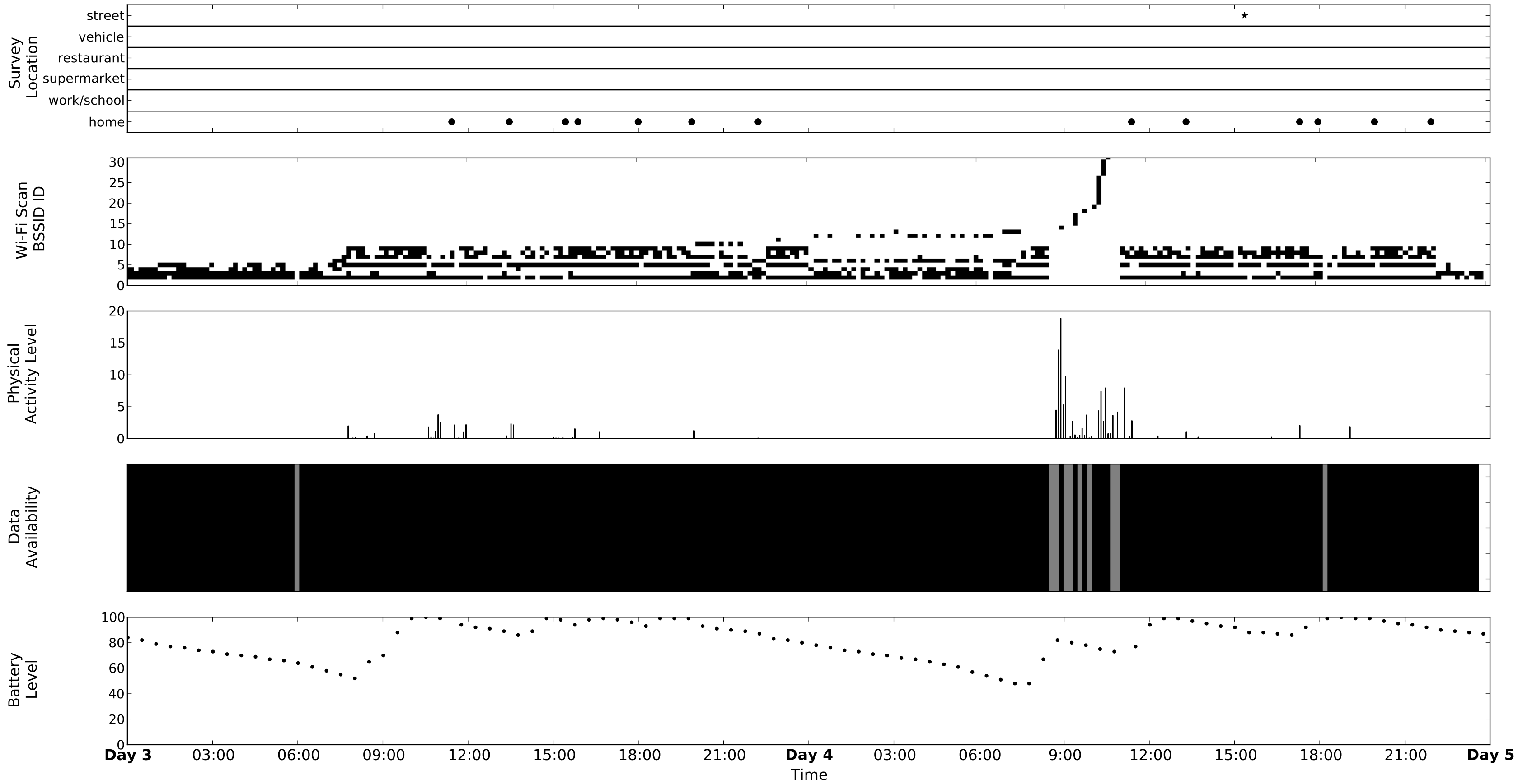


Figure 3.28. Multimodal analysis of the 15th participant's data. In most of the surveys, user mentioned his/her location as home. We see high physical activity between 9:00 and 12:00 in Day 4. We also observe that new BSSIDs are introduced during this timespan but the user did not submit any surveys mentioning his/her location during this timespan.

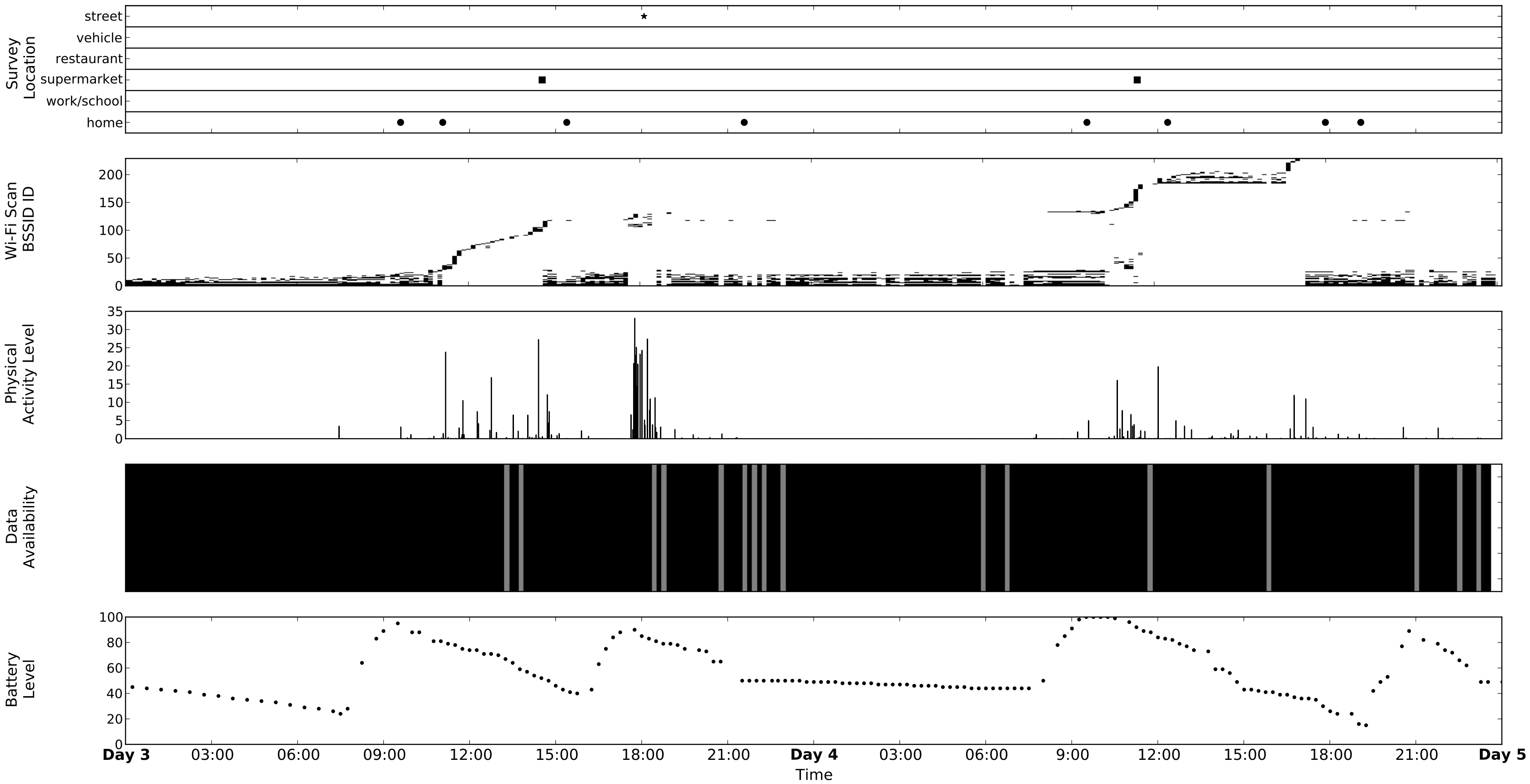


Figure 3.29. Multimodal analysis of the 18th participant's data. This participant's occupation is housewife. We observe that she is at supermarket around 14:00 in Day 3 and around 11:00 in Day 4. We observe high physical activity around these times. Again, the user submitted her location as street via a survey at 18:00 in Day 3 and we also observe high physical activity around that time.

4. SEMANTIC LOCATION AWARENESS

In this chapter, we propose a novel algorithm to learn semantically meaningful places of smartphone user. The algorithm periodically conducts Wi-Fi scans. When it detects a stable radio environment, it sends a notification to the user that requests giving a name to the current location. In this algorithm, we employ real time user feedback to learn both the names of the places and the Wi-Fi fingerprints of the places a user visits during a day.

4.1. Motivation

Places where a person goes and stays for a while can be considered as the significant places of the person. For example, Home, Office, and Parent's Home can be the names of some of the significant places of a person. Various algorithms are proposed in the literature to learn significant places of a person from the data gathered by a mobile device s/he uses. These algorithms can be classified into two categories namely geomery-based algorithms and fingerprint-based algorithms. Geometry-based algorithms mostly take as an input the location history of a person obtained from the GPS service of the mobile device. They cluster the location points which are latitude and longitude pairs using various types of clustering algorithms and fit polygons to determine the boundaries of the determined regions. The main problem of the clustering algorithms is that they require parameters such as number of clusters or maximum size of a cluster to process the given data. However, if there is no prior knowledge on the number of significant places in one's location history, it is not possible to supply these parameters to an algorithm. These parameters are generally determined experimentally and various heuristics are used. Fingerprint based algorithms take as input the data from pervasive beacons such as Wi-Fi, Bluetooth and GSM Cell Tower signals and unique identifiers associated with these beacons. For example, APs which provide Wi-Fi service broadcast their MAC addresses. Each GSM cell tower has a unique cell identifier associated with it. Then these algorithms generate fingerprints for the locations a person visits from the beacon data by looking at the existence of beacons in these locations. For example, when a previously known beacon is found in a location, the person is determined as in the place previously associated with the beacon.

In Table 4.1, the characteristics of the algorithms proposed in the literature for finding significant places of a mobile device user are summarized. These algorithms differ in the input data they take in. Some of them are capable of learning indoor places of a user. Some of them employ user feedback to learn the names of the found significant places. Finaly, it is important for an algorithm to be able to work on-line which means performing all the computations of the algorithm on a mobile device. In Section 2.2, the details of these algorithms are given.

Table 4.1. Summary of the significant location finding algorithms in the literature.

Paper	**Input Data**	**Indoor?**	**User Feedback?**	**Online?**
Marmasse *et al.* [25]	GPS	X	✓	✓
Ashbrook *et al.* [26]	GPS	X	X	X
Ye *et al.* [32]	GPS	X	X	X
Laasonen *et al.* [34]	GSM	X	X	✓
Montolu *et al.* [27]	GPS, Wi-Fi, Accelerometer, GSM, Bluetooth	X	X	X
Kang *et al.* [30]	GPS, Wi-Fi	X	X	✓
LaMarca *et al.* [33]	Wi-Fi, GSM, Bluetooth	X	X	✓
Hightower *et al.* [35]	Wi-Fi, GSM	✓	X	X
Kim *et al.* [36]	GPS, Wi-Fi	✓	X	✓
Kim *et al.* [37]	GPS, Wi-Fi, Accelerometer	✓	✓	✓
Our Algorithm	Wi-Fi	✓	✓	✓

None of the algorithms except our proposed algorithm given in Table 4.1 employ user feedback during learning process. However, some of them use the feedback after the execution of their algorithm to evaluate their algorithm's accuracy in determining the significant places of a user. However, in order to learn the semantically meaningful places of a mobile device user, user feedback is required. For example, a coffee shop may be a work place for a person. The same place can be a part of a morning routine for another person where s/he buys a cup of coffee. Furthermore, meaning of place may change for a person over time. For example, a person who works in the coffee shop may quit his/her job and start to buy coffee from the same shop regularly. In order to capture the meaning of a place for a person, an algorithm needs to incorporate the user feedback into its learning mechanisms.

In order to give semantic location awareness to Auto Diary mobile application which is introduced in Chapter 5, we developed an algorithm which employs real time user feedback to learn visited locations of a smartphone user. In this chapter, we give the details of this algorithm.

4.2. Methodology

In our algorithm we periodically perform Wi-Fi scans to determine nearby AP MAC addresses namely BSSIDs. We look for BSSIDs that are consistently found in consecutive Wi-Fi scans. We put these BSSIDs in a set named active BSSID set. If a BSSID in the active set is not found in the latest Wi-Fi scan it is removed from the active set. If any one of the BSSIDs in the active set is found in consecutive Wi-Fi scans for a significant time, we conclude that the smartphone is in a stationary state with respect to the location of at least one of the APs whose BSSID is in the active set. We say that a location is important for a person if s/he stays in a location more than a certain amount of time. For example, if we conduct Wi-Fi scans with two minutes intervals and set the scan count threshold, which is the minimum number of Wi-Fi scans to detect a stay event, as five, the places where the user stays at least ten minutes will be detected as important places. When a staying event is recognized, a name for the current location is requested from the user to remember the location in the future. A new place is created when the user supplies a name for his/her location. All of the BSSIDs in the active set are then associated with the newly created place. As a result of subsequent Wi-Fi scans, we determine the current place of the smartphone as follows. We determine the places which have previously been associated with at least one of the BSSIDs found in the latest Wi-Fi scan. If there is only one place that is associated with at least one of the BSSIDs in the latest Wi-Fi scan, we determine the location of the smartphone as this place. If there are more than one place, we determine the place which have the most similar Wi-Fi fingerprint to the latest Wi-Fi scan's fingerprint. The place with the most similar fingerprint is chosen as the current place of the user.

Definitions of the terms used in the explanation of the algorithm are as follows. BSSID is the MAC address of a wireless access point which uniquely identifies it. Active BSSID Set is a set of BSSIDs which always contains the BSSIDs found in the latest Wi-Fi scan. We also keep a scan counter for each of the BSSIDs in this set. Place BSSID Set is a set of BSSIDs associated with a place. $scanCountThreshold$ is the minimum number of scans required to determine that the smartphone is staying in a place. User Approved BSSID is a BSSID in Place BSSID Set which had a scan count greater than $scanCountThreshold$ at the time of place creation. We rely on this type of BSSIDs to determine the current place of a user. Persistent BSSID is a BSSID that is in the active BSSID set and has a scan count greater than $scanCountThreshold$. We say that a smartphone is in a stationary state if any one of the BSSIDs in its active BSSID set has a scan count greater than $scanCountThreshold$. Active Place is a place where the smartphone is currently staying at. Probable Place is a place which has at least one of the BSSIDs in the Active BSSID Set in its Place BSSID Set.

Flowchart of the proposed algorithm is given in Figure 4.1. Sub-procedures used in the flowchart are described in detail in the following sections. Sample execution of the algorithm under various scenarios are also investigated.

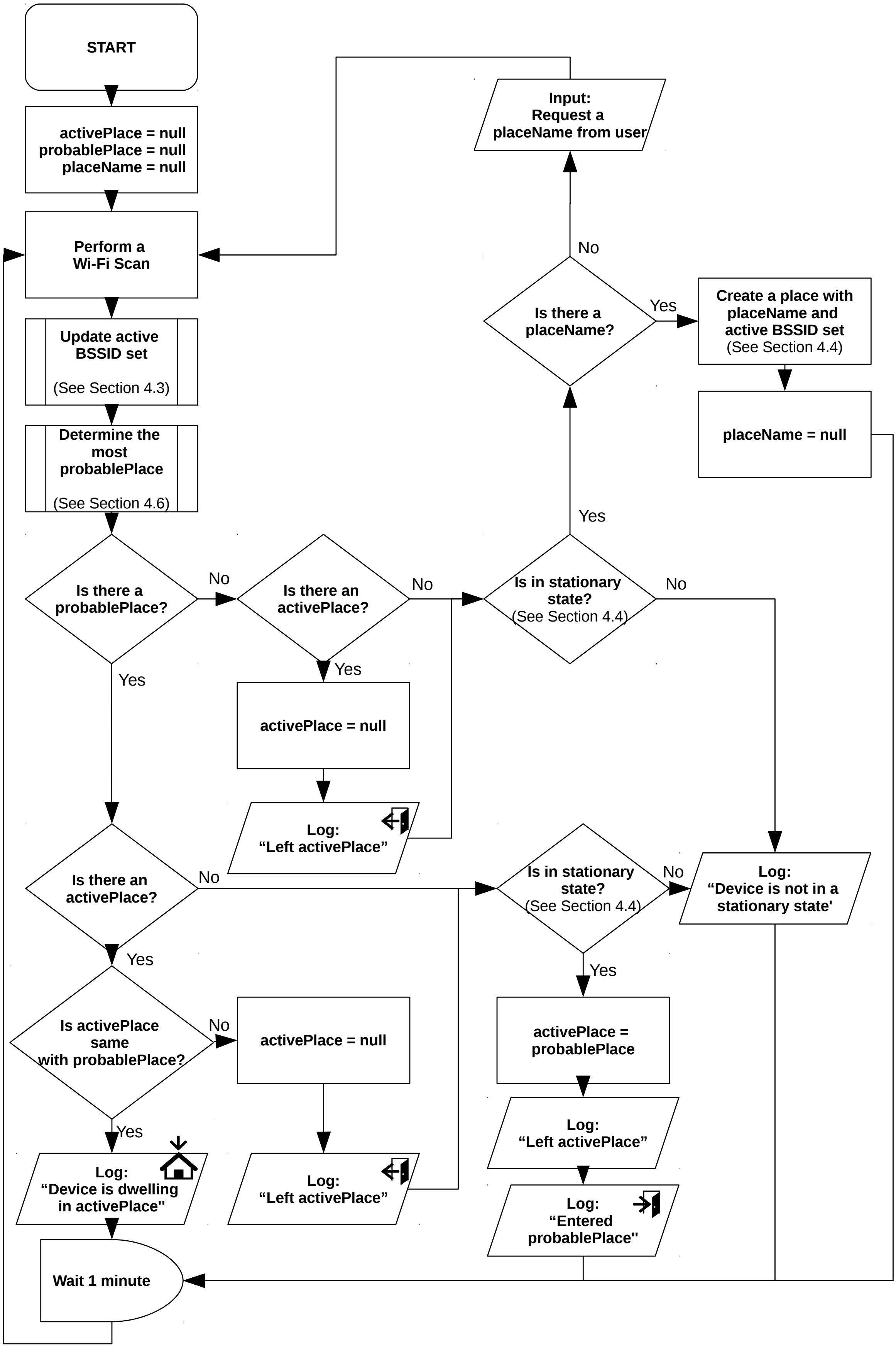


Figure 4.1. Flowchart of the proposed Wi-Fi localization algorithm.

4.3. Algorithm for Updating Active BSSID Set

Pseudo code of the algorithm which updates the active BSSID set is given in Figure 4.2. The active BSSID set contains only the BSSIDs that are found in the latest Wi-Fi scan. For each of the BSSIDs in the active set a scan counter counts the number of times a BSSID was seen in consecutive Wi-Fi scans. If a BSSID in the active set is not found in the latest Wi-Fi scan it is removed from the active set (line 3). If it is found in the scan its scan counter is incremented by one (line 11). If a new BSSID that is not already in the active set is found, it is added to the set and its scan counter is initialized to one (line 13). If the active set becomes empty after removing the BSSIDs that are not found in the latest Wi-Fi scan, we update the stationary state start time as current time (line 7).

Input: $wifiScanResultBssidSet$, $activeBssidScanCountDictionary$, $stationaryStateStartTime$

1: **for all** $bssid \in activeBssidScanCountDictionary.keys()$ **do**
2:   **if** $bssid \notin wifiScanResultBssidSet$ **then**
3:     $activeBssidScanCountDictionary.remove(bssid)$
4:   **end if**
5: **end for**
6: **if** $activeBssidScanCountDictionary.size() = 0$ **then**
7:   $stationaryStateStartTime \leftarrow currentTime()$
8: **end if**
9: **for all** $bssid \in wifiScanResultBssidSet$ **do**
10:   **if** $bssid \in activeBssidScanCountDictionary.keys()$ **then**
11:     $activeBssidScanCountDictionary[bssid] += 1$
12:   **else**
13:     $activeBssidScanCountDictionary[bssid] = 1$
14:   **end if**
15: **end for**

Figure 4.2. Algorithm which updates an active BSSID set after a Wi-Fi scan.

A sample execution of the algorithm which updates the active BSSID set is given in Table 4.2. Let A, B, C and D be four different BSSIDs. As a result of the first Wi-Fi scan at t_1, A, B and C are found. They are added to the active BSSID set and their scan counter is initialized to one. At t_2, C is not found in the Wi-Fi scan so it is removed from the active BSSID set. Scan counters of other BSSIDs are incremented by one. At t_n we see that A has a scan count of n which means it was found in all of the Wi-Fi scans until t_n. At t_{n+2}, A is not found in the scan and it is removed from the active BSSID set. At t_{n+2} a new BSSID named D is found. Between t_{n+2} and t_{2n+1}, D is found in all of the Wi-Fi scans.

In the next section, the algorithm which determines whether a smartphone is in a stationary state by looking at the BSSID scan counts is given.

Table 4.2. Sample execution of active BSSID set update algorithm. Scan count of a BSSID is incremented if it is found in the latest Wi-Fi scan. If it is not found, it is removed from the active BSSID set. Value of $scanCountThreshold$ is n.

Time	**Stationary State Start Time**	**Wi-Fi Scan Result BSSID Set**	**Active BSSID:ScanCount Dictionary**	**Is in Stationary State?**
t_1	t_1	$\{A, B, C\}$	$\{A : 1, B : 1, C : 1\}$	no
t_2	t_1	$\{A, B\}$	$\{A : 2, B : 2\}$	no
t_3	t_1	$\{A, C\}$	$\{A : 3, C : 1\}$	no
...	...	...	...	...
t_n	t_1	$\{A, B, C\}$	$\{A : n, B : m, C : k\}$	yes
t_{n+1}	t_1	$\{A, B\}$	$\{A : n + 1, B : m + 1\}$	yes
t_{n+2}	t_{n+2}	$\{D, C\}$	$\{D : 1, C : 1\}$	no
t_{n+3}	t_{n+2}	$\{D\}$	$\{D : 2\}$	no
...	...	...	...	no
t_{2n+1}	t_{n+2}	$\{D\}$	$\{D : n, C : j\}$	yes
...	...	...	...	...

4.4. Determining the Stationary State and New Place Creation

We say that a smartphone is in stationary state if any one of the BSSIDs in its active BSSID set has a scan count greater than or equal to the minimum scan count required for recognizing a staying event namely *scanCountThresholdForStay*. The algorithm that determines whether a smartphone is in a stationary state is given in Figure 4.3. In Table 4.2, the smartphone is in stationary state at t_n, t_{n+1} and t_{2n+1}.

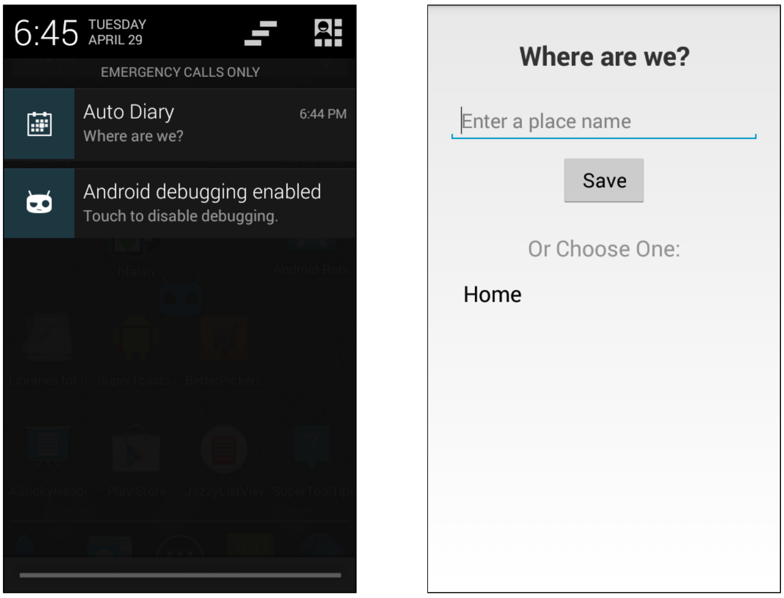
Input: $activeBssidScanCountDictionary$, $scanCountThreshold$

Output: **true** or **false** {Is in stationary state?}

```
1: for all scanCount ∈ activeBssidScanCountDictionary.values() do
2:    if scanCount ≥ scanCountThreshold then
3:       return true
4:    end if
5: end for
6: return false
```

Figure 4.3. Algorithm which determines whether a smartphone is in stationary state after a Wi-Fi scan.

When a smartphone is determined as in stationary state, a notification is sent to the user of the smartphone requesting a place name. Figure 4.4 shows a sample notification and the user interface that is used for naming a place. When the notification in Figure 4.4a is tapped, the screen in Figure 4.4b is shown to the user. After a name is supplied by the user using this screen, a new Wi-Fi scan is conducted immediately. If the smartphone is still in a stationary state after the latest Wi-Fi scan, all of the BSSIDs found in the latest Wi-Fi scan are associated with the given name and a new place is created. If the smartphone is not in a stationary state after the latest Wi-Fi scan, the given name is not used and a new place is not created.



(a) (b)

Figure 4.4. The user interfaces used for naming places: (a) a sample notification requesting a place name (b) The user interface that is used for naming a place.

4.5. Learning the BSSID Set of a Place

In Figure 4.5, a hypothetical AP toplogy is shown for an office building. Let A, B, C, D and E be five APs in this environment. The circles around the APs show the signal range of them. The scenario described in Table 4.2 can be valid for this topology. Consider a smartphone staying inside the signal range of A on the second floor of the building. Since the smartphone is in the range of A, it is expected that A will be found in consecutive Wi-Fi scans consistently given that the smartphone stays in the range. However, B and C may not be found in some of the periodic Wi-Fi scans due to their weak signals as they are located on different floors. When the scan count of A reaches *scanCountThreshold*, a notification will be sent to the user of the smartphone asking for a place name for this location. The user will supply a name such as “Office” using the screen shown in 4.4b. After a name is given by the user, a new Wi-Fi scan is conducted. If the smartphone is still in a stationary state, a new place is created with the given name. All of the BSSIDs that are found in the latest Wi-Fi scan are added to the BSSID set of the newly created place. We say that, at the time of place creation, BSSIDs that have a scan count greater than or equal to *scanCountThreshold* are the user approved BSSIDs of a place. That means A will be the only user approved BSSID of the newly created place in this scenario given that it is the only BSSID consistently found in consecutive Wi-Fi scans. B and C may also be found in the Wi-Fi scan conducted just before the place creation. Suppose that C is also found in the Wi-Fi scan conducted just before the place creation. It is also added to the BSSID set of the newly created place which is named as “Office” by the user. However, it is not tagged as a user approved BSSID. Adding C to the place's BSSID set allows us to merge “Office” with other places where the signal of C reaches inside the building. This mechanism will be explained next.

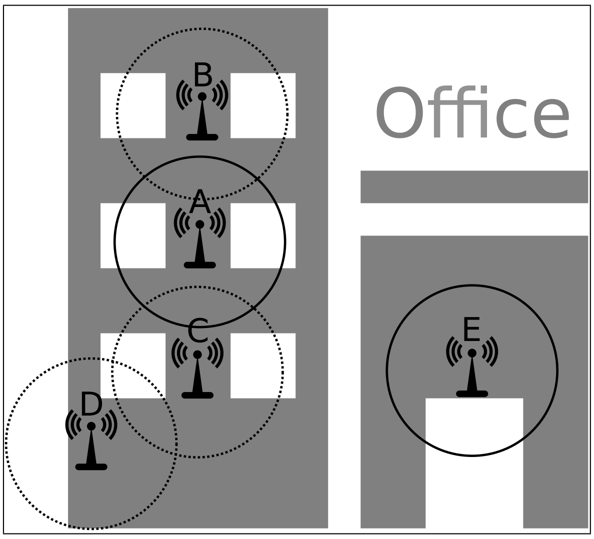


Figure 4.5. A hypothetical AP topology for an office environment. APs that have a user approved BSSID are shown with a solid circle around them. Other BSSIDs are shown with a dashed circle around them.

Now consider that the user moved to the first floor of the building and is now located inside the signal range of D. Suppose that as a result of the periodic Wi-Fi scans only D and C are found in this new place. When the smartphone reaches to the stationary state, a new notification will be sent to the user asking for a place name. However, this time C, which is already in one of the previously created place's BSSID set, is found in the latest Wi-Fi scan. In the place naming screen which is shown in Figure 4.4b, “Office” will be shown under the “Or Choose One:” section. This implementation allows to merge indoor places which share at least one BSSID. In this scenario, C is the only shared BSSID between the two places. If the user taps on a name of one of the previously created places in Figure 4.4b, all of the BSSIDs found in the latest Wi-Fi scan are added to the BSSID set of the tapped place. If the user chooses to give a new name to the place instead of merging it with a previously created place, a new place is created with the BSSIDs found in the scan as it was described before.

As a final case, the user can now move to a location inside the signal range of E. Since there is no AP nearby that is shared with a previously created place, the user has to give a new name to this location. In the place naming screen shown in Figure 4.4b, no place name will be shown for merging when a place name is requested for this location.

4.6. Detecting Entrance and Departure Events

After each Wi-Fi scan we determine the current place of the smartphone. We first determine the BSSIDs in the active BSSID set which have a scan count greater than *scanCountThreshold*. We call these BSSIDs as *persistent BSSIDs*. For example, if the value of *scanCountThreshold* is five, a BSSID should be found in five consecutive Wi-Fi scans in order we to consider it as a persistent BSSID. Persistent BSSIDs are the ones that we rely on in determining the current place of the user. BSSIDs with a scan count less than *scanCountThreshold* may be temporarily in the active BSSID set. After determining the persistent BSSIDs, we determine the places which were previously associated at the time of place creation with at least one of the persistent BSSIDs as one of their user approved BSSIDs. One of these places will be determined as the current place of the user. If there is no place associated with at least one of the persistent BSSIDs, a notification is sent to the user requesting a name for the current location in order to create a new place. If there is only one place, then this place is determined as the current place of the user. If there are more than one candidate places for the current location of the user, we determine the place which have the most similar Wi-Fi fingerprint to the current location's Wi-Fi fingerprint. A Wi-Fi fingerprint is a bit string in which each bit denotes the existence of a BSSID in the fingerprint of the place. In Table 4.3, three sample Wi-Fi fingerprints are given. For example, $BSSID_1$, $BSSID_2$ and $BSSID_3$ exist in the Wi-Fi scan result's fingerprint whereas only $BSSID_3$ exists in the fingerprint of Place B. We use Tanimoto similarity to measure the similarity of each candidate places' Wi-Fi fingerprint to the Wi-Fi scan result's Wi-Fi fingerprint [55]. The definition of Tanimoto similarity is given in Equation 4.1. In this equation, X and Y are bit strings and X_i denotes the value of ith bit. $\wedge$, $\vee$ are bitwise and, or operators respectively. The place which has the fingerprint most similar to the Wi-Fi scan result's fingerprint is determined as the current place of the user. For example, in Table 4.3, Place A will be determined as the current place of the user with a similarity score of 0.5. Place B's similarity score is determined as 0.33. Since the similarity of score of Place A's fingerprint is higher, Place A is determined as the current place of the current place of the smartphone. Most of the time, similarity calculations are not needed if the user chooses to merge nearby places. For example, in Figure 4.5, a user may merge all of the places inside the building and give a single name to whole building given that signals of APs cover the entire building.

Table 4.3. Wi-Fi fingerprint generation. A 1 denotes the existence and 0 denotes the absence of the corresponding BSSID in the fingerprint of a place. For example, only $BSSID_3$ exits in the Wi-Fi fingerprint of Place B.

	BSSID$_1$	**BSSID$_2$**	**BSSID$_3$**	**BSSID$_4$**
Wi-Fi scan result	1	1	1	0
Place A	1	1	0	1
Place B	0	0	1	0

$$T_b(X,Y) = \frac{\sum_i (X_i \wedge Y_i)}{\sum_i (X_i \vee Y_i)} \tag{4.1}$$

Departure events can occur in two ways. First, if as a result of a Wi-Fi scan there is no probable place that has one of the BSSIDs found in the scan in its BSSID set then a departure from the current active place is logged. Second, if the current active place is different from the most probable place determined as a result of the latest Wi-Fi scan, a departure from the current active place is logged.

4.7. Sample Executions of the Algorithm

Sample execution of the algorithm for over 21 hours is illustrated in Figure 4.6. The period of the Wi-Fi scans is set as two minutes. $scanCountThreshold$ is set as three. That means if the smartphone spends at least six minutes in a location, the smartphone enters into stationary state. A unique ID is given to each BSSID found in the Wi-Fi scans during this period. Left axis of the figure shows the unique IDs of the BSSIDs. A black region on the row of a BSSID at a given time implies that the BSSID is found in the scan performed at that time. Dashed lines show the times when a place change is recognized by the algorithm. Figure 4.7 shows the zoomed view of the beginning hours of Figure 4.6. In this figure, we also show the descriptions of the events recognized by the algorithm. For example, we observe that user left Office at 21:52 and entered Home at 22:30. Figure 4.8 shows the zoomed view of the last hours of Figure 4.6. For example, we observe that the user left Home at 09:14 and entered Office at 09:20. Office, Home and ETA Table Tennis are colloquial names the user chose to name his/her places during place creation. The algorithm is able to capture the place changes without a delay thanks to $stationaryStateStartTime$ updated in line 7 of Figure 4.2. The algorithm uses the $stationaryStateStartTime$ as the entrance time when it is sure that it is in stationary state after three consecutive Wi-Fi scans. If $stationaryStateStartTime$ were not used, the algorithm would experience a six minutes delay in logging an entrance event in this execution.

In Figure 4.9, sample execution of the algorithm for a different user is shown. The period of the Wi-Fi scans is set as two minutes. Dashed lines show the times when a place change is recognized by the algorithm. For example, the user entered supermarket at 14:32 and left supermarket at 14:36.

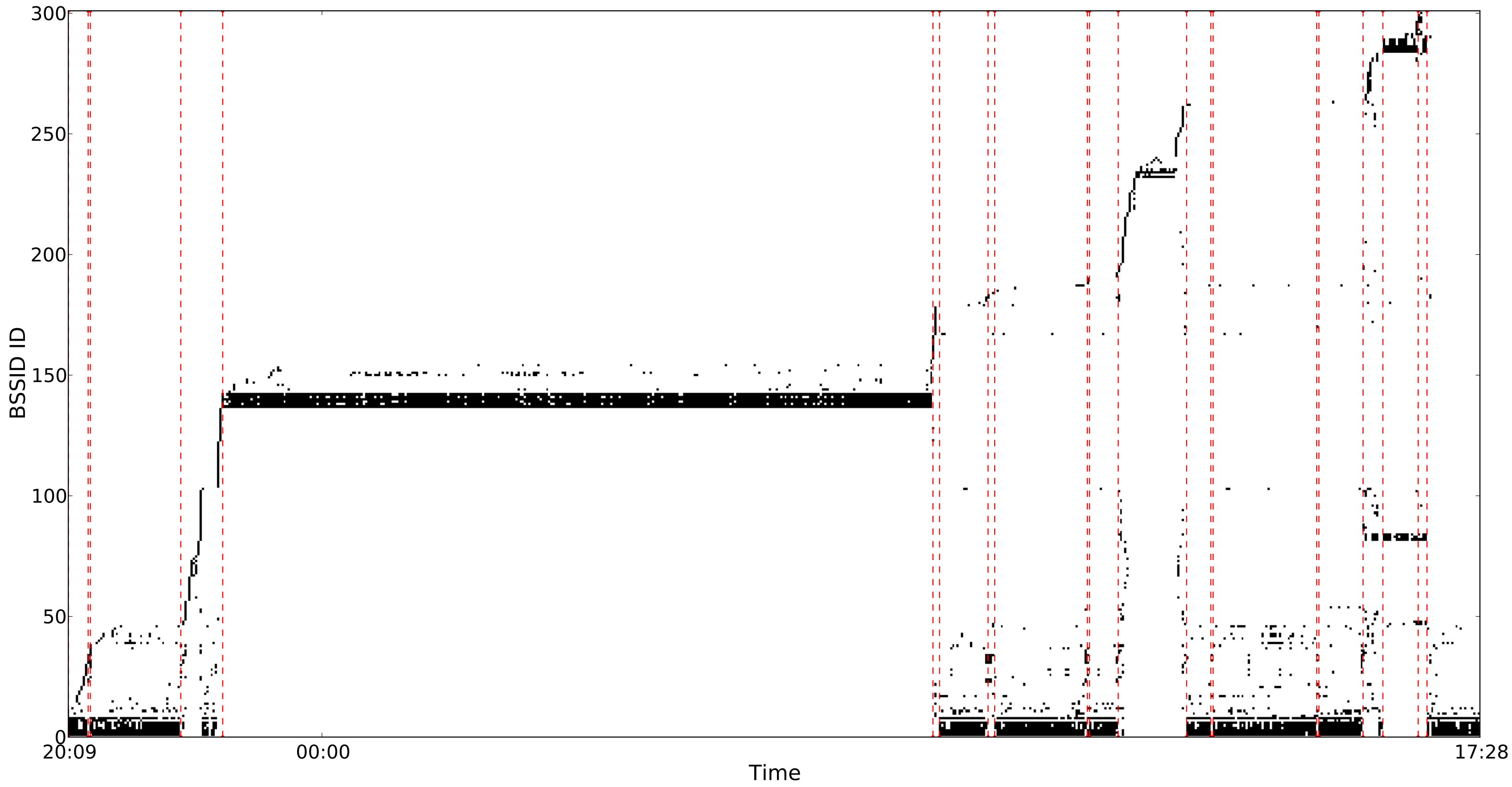


Figure 4.6. Sample execution of the algorithm for over 21 hours. A unique ID is given to each BSSID found in the Wi-Fi scans during this period as it is shown on the left axis. A black region on the row of a BSSID at a given time implies that the BSSID is found in the scan performed at that time. Dashed lines show the times when a place change is recognized by the algorithm.

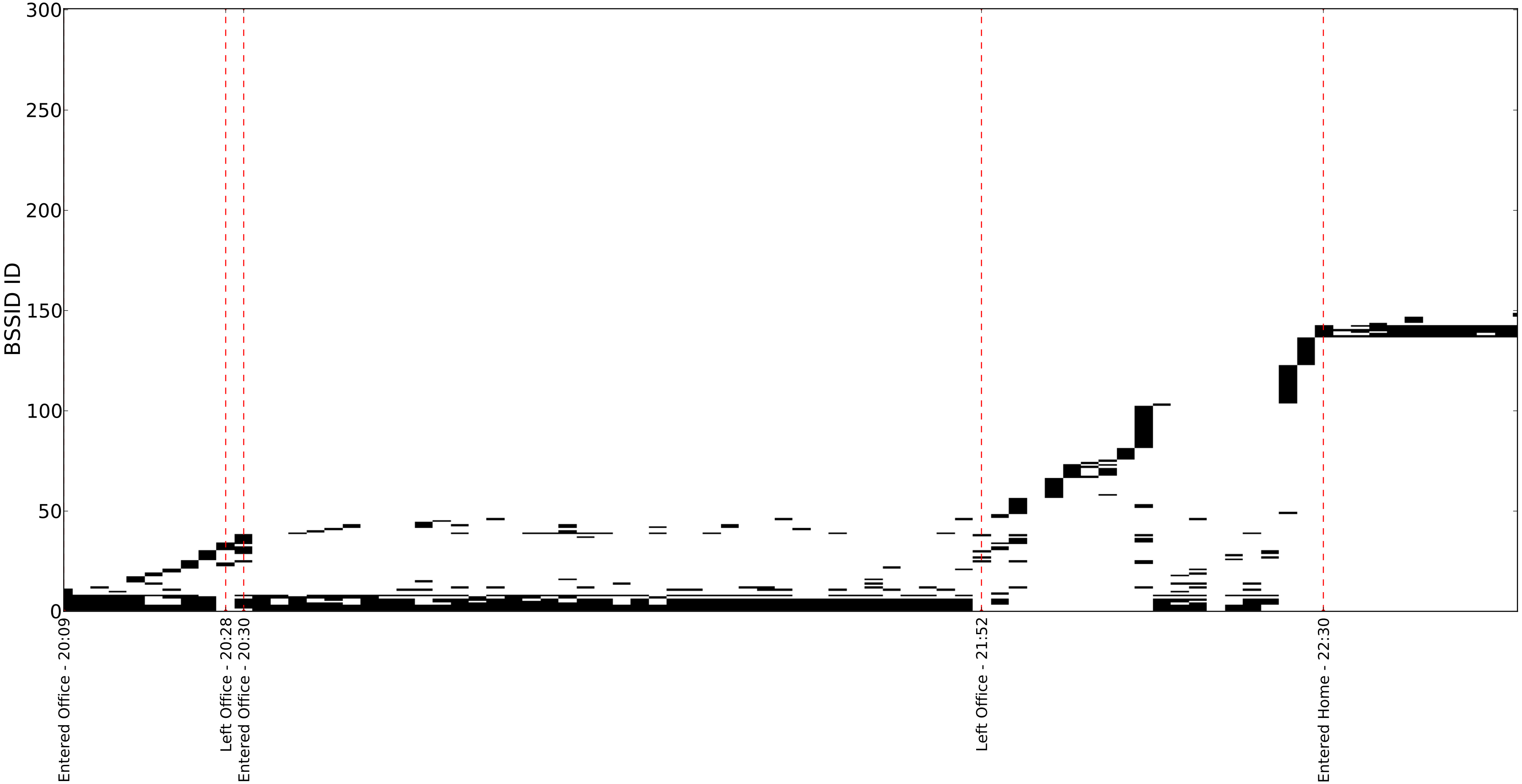


Figure 4.7. Zoomed view of the first hours of Figure 4.6. Place entrance and departure events are shown with dashed lines. For example, the user left Office at 21:52 and entered Home at 22:30.

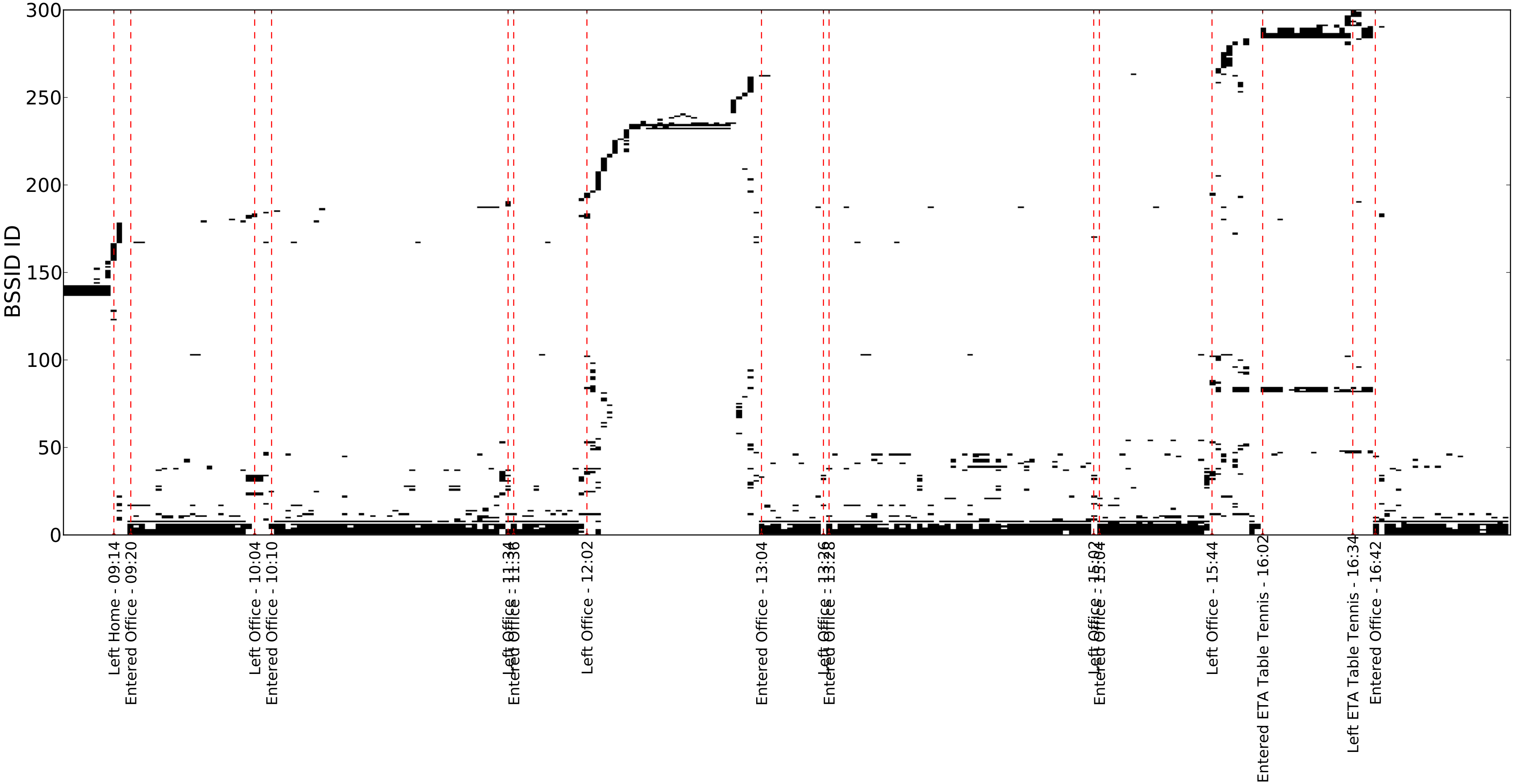


Figure 4.8. Zoomed view of the last hours of Figure 4.6. Place entrance and departure events are shown with dashed lines. For example, the user entered a place with a colloquial name "ETA Table Tennis" at 16:02.

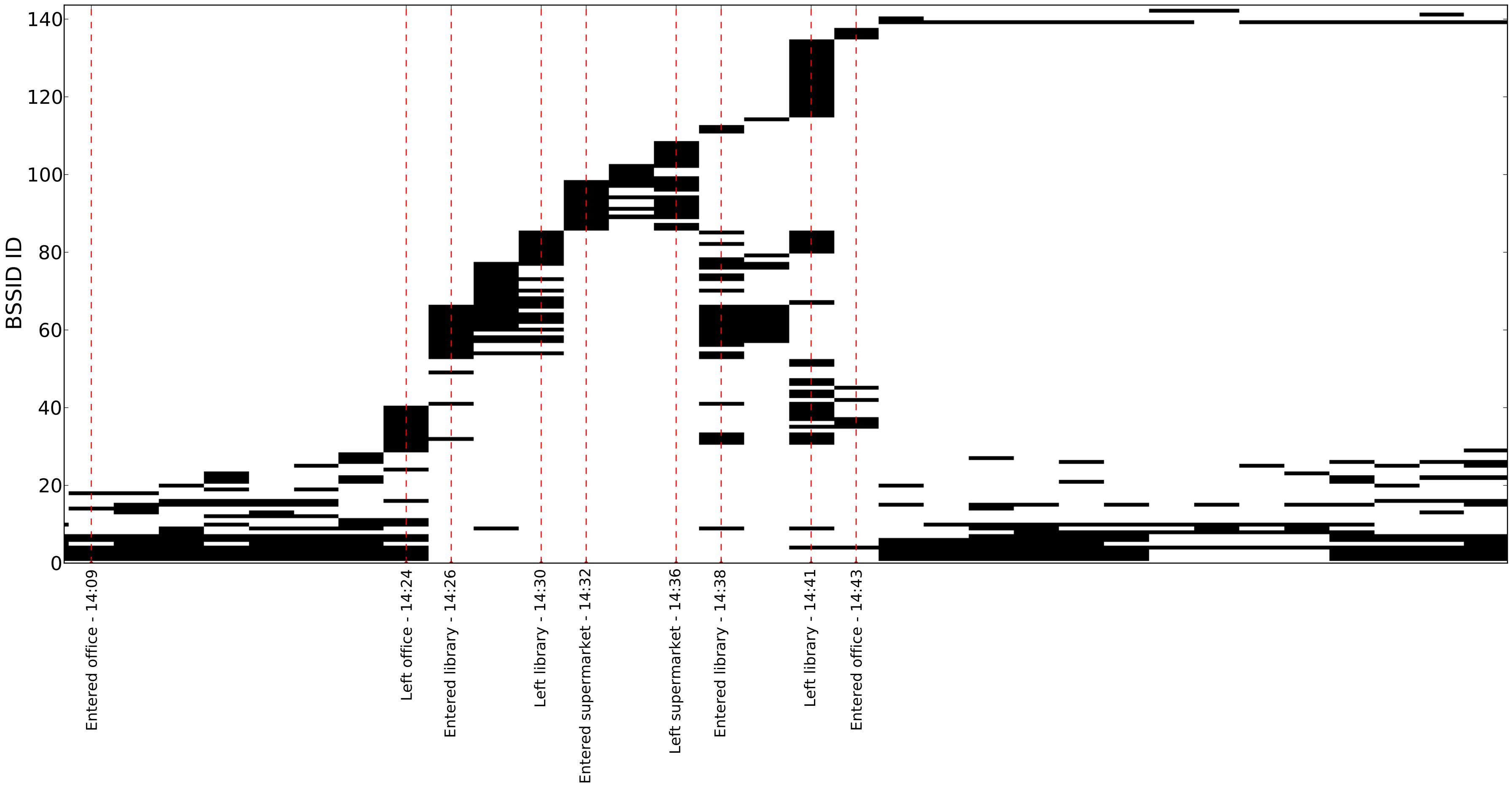


Figure 4.9. Sample execution of the localization algorithm for an hour. Place entrance and departure events are shown with dashed lines. For example, the user entered supermarket at 14:30 and left supermarket at 14:36.

5. AUTO DIARY

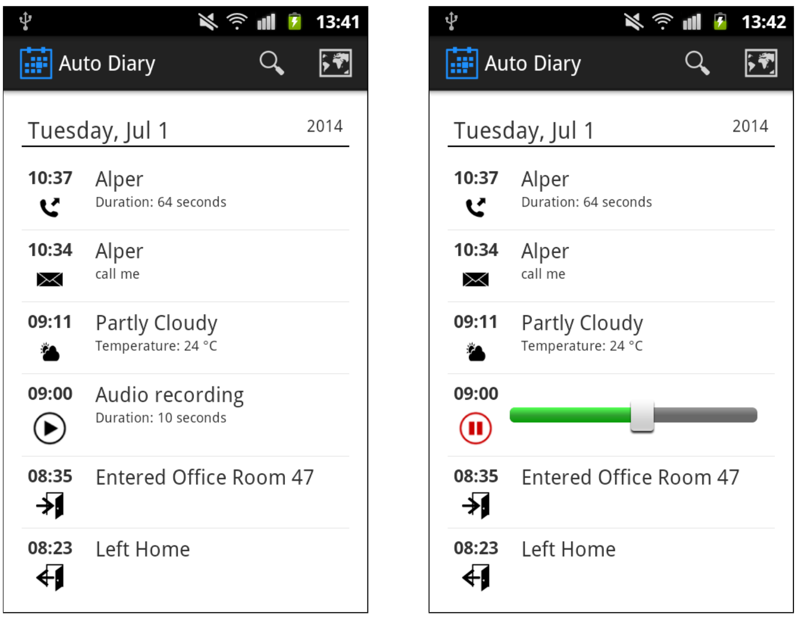
In this chapter, we present a mobile lifelogging application named Auto Diary. This application is targeted for mobile devices running Android OS. In the first section, motivation of the application is given. In the following sections, implementation details of the application are given.

5.1. Motivation

Our digital fingerprints are scattered among many mobile applications. For example, we make phone calls and a call log application keeps the history of the calls we have made. We use SMS and an SMS application keeps the messaging history. A weather application displays today's weather. Furthermore, location-based services (LBS) track the location of a smartphone in order to offer location aware services such as recommending nearby social events [56]. However, there is no mechanism that allows a smartphone user to efficiently use his/her digital fingerprints as cues to retrieve his/her past experiences. For example, imagine a smartphone user who tries to find a specific SMS message inside an SMS application. She remembers that it was a snowy day and she received the message on a Sunday morning at home. She also remembers that she received a phone call from her mother on the same day. If there are too many messages in her inbox, she is hopeless in this scenario and should take a look at all of the SMS messages in the inbox to find the right one. However, if she were able to specify the cues that she remember from the day she received the message to a context-aware messaging application, she would be able to efficiently retrieve the content of the message in a short time period. This is just an example scenario. All of the services of a smartphone, not just the SMS messaging application, can benefit from context-awareness and offer appropriate retrieval functionality. In the remaining of this chapter, we present a context-aware mobile lifelogging application named Auto Diary. This application records call logs, SMS messages, weather conditions, visited locations and ambient audio periodically. Visited locations are logged using the algorithm introduced in Chapter 4. All of the captured experiences are displayed in a daily timeline to the user of a smartphone. In addition, a retrieval functionality is offered which allows to specify cues to find a day with a matching description.

5.2. Data Collection

Auto Diary automatically records visited locations, weather conditions, SMS messages, call logs and ambient audio. All of the data are stored in an SQLite database except the audio recordings. The name of audio files are stored in the SQLite database and the raw audio files are stored in the file system of the Android OS. All of the captured data are shown in a daily timeline to the user. Figure 5.1 shows a sample timeline from the application. At 08:23, the user left Home and entered Office Room 47 at 08:35. Ten seconds long audio is recorded at 09:00. A user is able to listen the recorded audio files as it is shown in Figure 5.1b. A weather report for 09:11 is received from an online weather service. At 10:34, an SMS message with “call me” as its content is received from one of the contacts of the user named Alper. The user called the sender of the message three minutes later at 10:37 and talked for 64 seconds.



(a) (b)

Figure 5.1. A sample timeline from Auto Diary: (a) timeline (b) audio is playing.

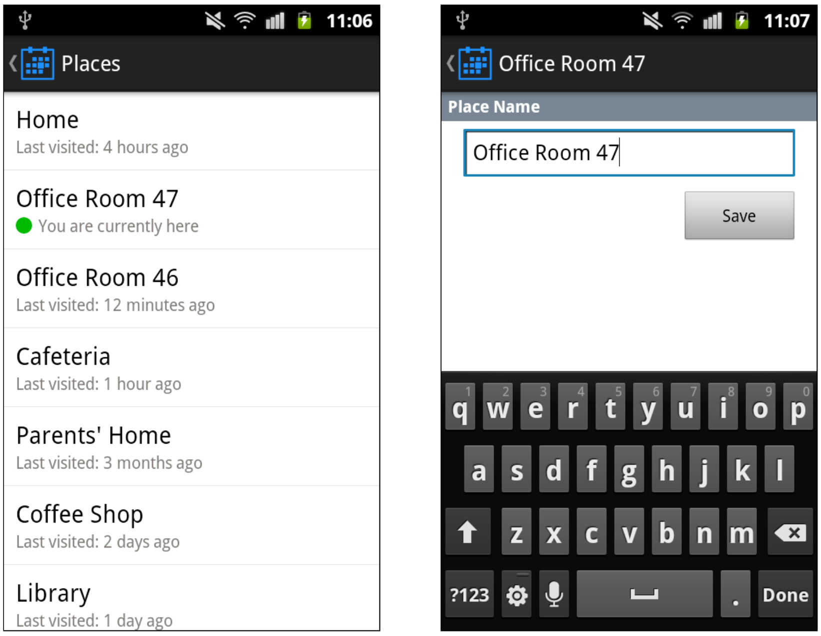
In the remaining of this section, details of the data collection are given for each of the data modalities that are shown on the timeline.

5.2.1. Visited Locations

We use the algorithm introduced in Chapter 4 to learn the semantically meaningful places of a smartphone user and then track visits to these places. A summary of this algorithm is as follows. The algorithm periodically conducts Wi-Fi scans. If an AP is found in consecutive Wi-Fi scans for a significant amount of time that means the user is in a stable radio environment. When a stable radio environment is found, the user of a smartphone is notified to give a name to his/her current location (Figure 4.4). When a name is supplied by the user, all of the APs found nearby are associated with the given name and a new place is created. Later, when a new Wi-Fi scan is conducted, the algorithm compares the live Wi-Fi fingerprint in the location with the fingerprints of previously created places. The place with the most similar fingerprint is logged as the current place of the user. In Figure 5.2, place management related user interfaces of Auto Diary application are shown. In Figure 5.2a, all of the previously created places are shown. Last visit times are shown under each of these places. Departure time is used as the last visit time to a place. For the place that the user is currently located at, a green dot and “You are currently here” message are displayed under the place name. A user can change the name of a place at any time using the interface shown in Figure 5.2b.

5.2.2. Weather Conditions

Weather conditions of the area that the user is currently located at are retrieved from an online weather service. We use the weather service provided by World Weather Online ® [57]. The service allows to specify the latitude and longitude coordinate values of the area that the weather conditions will be reported. We obtain the last known location of a smartphone in terms of latitude and longitude coordinates from the location API of Android OS. We supply this location to the weather service to get the weather conditions for the current location of the smartphone. There are 48 different weather condition types that may be reported by the service. “Clear/Sunny”, “Partly Cloudy”, “Heavy snow”, “Heavy rain”, “Fog” and “Heavy freezing drizzle” are examples of weather condition types that the service may report. Full list of the weather



(a) (b)

Figure 5.2. Place management related user interfaces in Auto Diary application: (a) previously created places (b) renaming a place.

conditions can be found in [58]. Internet connection is required to get the weather conditions from the web service. We periodically request weather conditions from the service. The response of the service contains weather condition type, temperature and the time that the weather conditions are observed by the weather service. Response data are stored in the SQLite database.

5.2.3. SMS and Call Logs

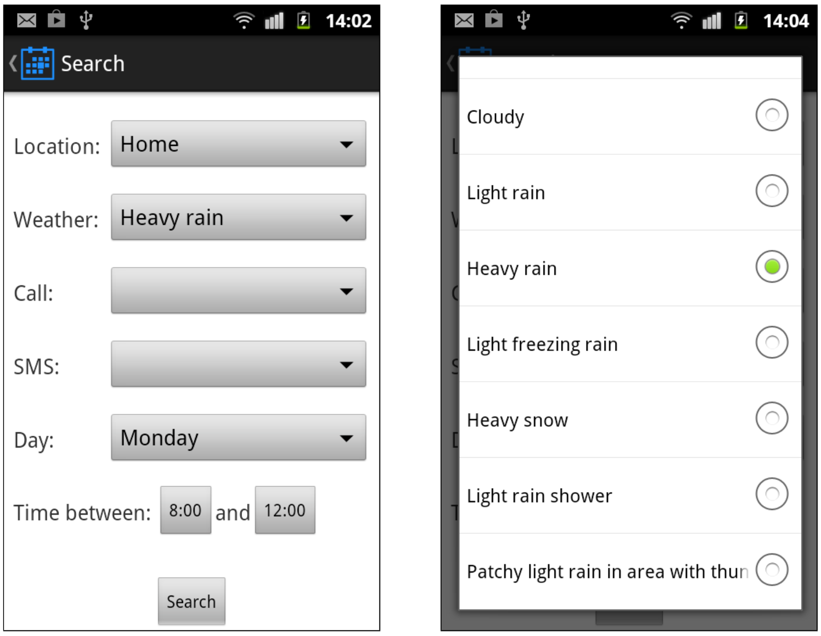
Android OS gives the SMS messages and call logs of a smartphone user when requested through related APIs. We request the call logs and sms messages of a smartphone user at each time the user opens Auto Diary application. The API also allows to specify the beginning time of the logs. We store the current time when we make a request and specify the stored time as the beginning time of the logs for the subsequent request. This way we shorten the time required to fetch the logs. For an SMS message, we store the name and phone number of the contact the message is sent to or received from, date of the message, content of the message, and type of the message (i.e., incoming or outgoing) in the SQLite database. For call logs, we store the name and phone number of the contact involved in a call, date and duration of the call, and type of the call (i.e., incoming, outgoing or missed).

5.2.4. Audio Recording

Audio is recorded periodically. The raw audio files are stored in the file system of Android OS. We use the timestamp of an audio recording as the file name of the raw audio. The name of the audio files are stored in the SQLite database. Currently, we record audio in duty cycles. For example, we record ten seconds of audio with one hour intervals. We envision that in the near future, through advances in sensor and battery technologies, full day continuous audio recording will be feasible using smartphones.

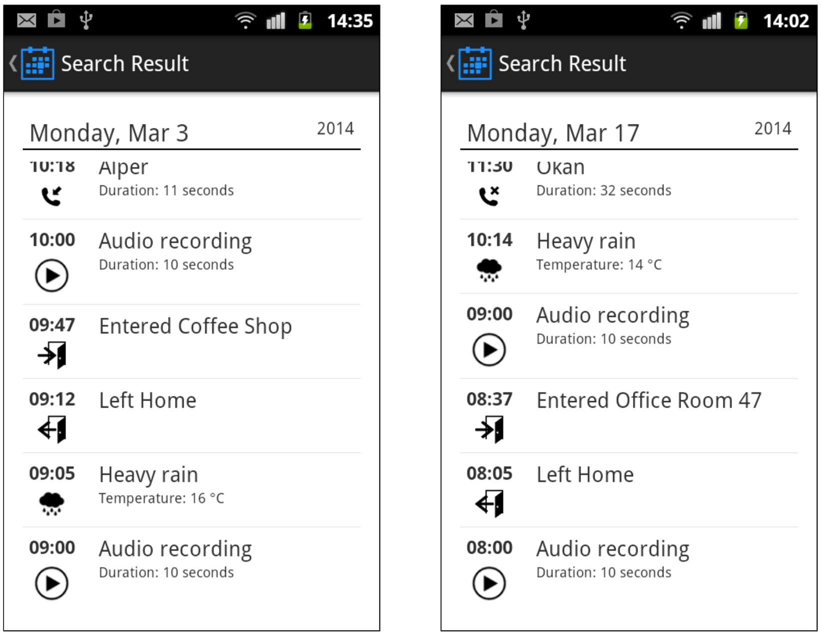
5.3. Experience Retrieval

We implemented an experience retrieval mechanism in Auto Diary that will allow a user to retrieve his/her experiences using associative cues. In Figure 5.3, experience retrieval user interfaces are shown. A user is able to specify cues from each of the modalities shown in Figure 5.3a to retrieve his/her past experiences. Location name and weather condition type can be specified as cues. In addition name of the day and timespan of the remembered experiences can be specified. For SMS and call events, a contact name can be specified. For example, in this case the user specified the cues from the sentence “It was a rainy Monday morning. I was at home”. In Figure 5.3b, all of the weather conditions that the user can specify are shown in a list. The content of each select list in the search screen is generated from the previous experiences of the user. Thus, the user has experienced each of the shown weather conditions at least one time before. When the user taps the search button on the search screen shown in Figure 5.3a, all of the days that matches with the query are retrieved. In Figure 5.4, two days are shown that matches with the query and the timelines of these days are shown to the user as a result of the search. The user is able to listen the audio files recorded on these days.



(a) (b)

Figure 5.3. Experience retrieval mechanism user interfaces: (a) select lists for specifying cues in different modalities (b) one of the weather condition types that the user has experienced before can be chosen as a weather cue.



(a) (b)

Figure 5.4. Result of the search given in Figure 5.3a: (a) first day that matches with the specified cues (b) second day that matches with the specified cues.

6. CONCLUSION

In this thesis, we first introduced a system named Smartphone Tracker which allows us to conduct large-scale data collection studies using Android smartphones. As a part of this system, we introduced a survey mechanism, a messaging infrastructure and a privacy preserving audio recording mechanism in which a user is able to review automatically recorded audio files before they are uploaded to server. Using this system, we conducted a five days long data collection study with 22 participants in collaboration with a researcher from the Department of Psychology at University of Zurich. Our collaborator designed surveys to get a comprehensive picture of the participants' daily activities, social interactions and personal thoughts. The analysis of the survey submissions and the audio recordings will be performed in this direction by our collaborator. We analyzed the collected data in order to obtain insights on the smartphone usage patterns of the participants and hardware differences between different smartphone models. We found that there are certain smartphone models from which we can collect various types of data reliably. We also experienced problems in data collection with certain smartphone models. We show high availability of Wi-Fi infrastructure for two housewives and a retired person who spent most of their time at home during the study. Televisions with Bluetooth technology are found consistently in Bluetooth scans of a few participants. We show through data completeness analysis, our data collection application may not collect data reliably and continuously from a smartphone due to various reasons such as empty battery, intentional shutdown of the smartphone by the user and Android OS energy optimization mechanisms that interfere with the data collection. In the multi-modal analysis of the collected data, we show that we are able to capture the place changes of a smartphone user both in accelerometer data and Wi-Fi data. Using survey answers, we verified that both of these modalities are accurate estimators of a smartphone user's place changes.

Through data collection study, we showed that presence patterns of APs accurately reflect place changes of a smartphone user. We also showed the high availability of Wi-Fi infrastructure for most of the participants during a day. Building upon this findings, we developed an algorithm that gives semantic location awareness to mobile devices. This algorithm periodically conducts Wi-Fi scans and when it detects a stable radio environment it requests a place name from the user. When a name is supplied by the user, Wi-Fi fingerprint of the location is associated with the given name. APs found in a location constitute the Wi-Fi fingerprint of that location. Visits to previously created places are detected by comparing the Wi-Fi fingerprint in a location with the previously created places' Wi-Fi fingerprints. The place with the most similar fingerprint is determined as the current place of the user. We are also able to merge places with different Wi-Fi fingerprints but sharing at least one AP. Through this feature, for example, a three floor building can be named as "Office" by merging the Wi-Fi fingerprints of the floors.

Finally, we introduced a mobile lifelogging application which records call logs, SMS messages, weather conditions, visited locations and ambient audio periodically. Visited locations are logged using the novel Wi-Fi based algorithm that we have introduced. All of the captured experiences are displayed in a daily timeline to the user. An experience retrieval functionality allows to retrieve past experiences using associative cues. For example, a user remembering a snowy Monday night in which s/he received phone calls and SMS messages from his/her friends can use these cues to retrieve the experiences in the day described by the cues. An appropriate retrieval user interface is designed in which a user can specify cues to find a day with a matching description. As a result of a search, all of the experiences captured in the matching day are shown to the user. A user is also able to listen the automatically recorded audio files on the daily timeline. Currently, audio is recorded in duty cycles. In the near future, we envision that full day audio recording will be feasible using smartphones.

As a future work, we plan to conduct extensive user studies to assess the reliability of the semantic location awareness algorithm. We also plan to put the Auto Diary application on the Android applications distribution platform and get feedback from the users. We may consider designing a user study in which we will be able to show the usefulness of Auto Diary application as a memory aid. We will also improve Smartphone Tracker system and conduct new data collection studies using it.

APPENDIX A: 5 DAYS 22 PEOPLE DATA COLLECTION STUDY

A.1. Participant Consent Form

Table A.1. Participant consent form in Turkish.

KATILIMCI BİLGİLENDİRME FORMU

'Günlük Hayatta Doğal Düşünce Süreçlerinin İncelenmesi'

Dr. Burcu Demiray Batur, Zürih Üniversitesi, Psikoloji Bölümü
Dr. Bert Arnrich, Boğaziçi Üniversitesi, Bilgisayar Mühendisliği Bölümü

Aşağıdaki bilgileri okumanız, bu araştırmanın içeriği ve doğası hakkında bilgi sahibi olmanız açısından önemlidir.

Araştırmanın amacı:
Bu pilot çalışmanın amacı, insanların günlük hayatlarında ve doğal ortamlarında neler hakkında düşündüklerini ve konuştuklarını incelemektir.

Katılım kriterleri:
Bu çalışmaya katılmak için (1) en az 18 yaşında olmalısınız, (2) anadiliniz Türkçe olmalı ve (3) Android sistemiyle çalışan bir akıllı telefona sahip olmalısınız.

Çalışma kapsamında yapmanız gerekenler:

- Bilgilendirme toplantısı (27 Mart 2014): Araştırmanın prosedürleri ve katılımcıların yapması gerekenler anlatılacaktır. Kullanılacak akıllı telefon uygulamasından örnekler gösterilecektir.
- Akıllı telefon ile veri toplama süreci (29 Mart – 2 Nisan): Akıllı telefon uygulamamız katılımcıların günlük hayatlarında arka planda çalışarak data toplayacaktır. Ayrıca, günde ortalama 6 kez sinyal vererek katılımcılardan kısa bir anket doldurmalarını isteyecektir. Bu anketi doldurmak yaklaşık bir dakika sürmektedir.
- Geribildirim toplantısı (4 Nisan): Katılımcılardan çalışma ile ilgili görüşleri, soruları ve geribildirimleri alınacaktır. Son olarak, katılımcılara yardımları için ödeme yapılacak ve ilgilerini çekiyorsa çalışmanın hipotezleri hakkında bilgi verilecektir.

Akıllı telefon uygulaması ve katılımcı gizliliği
Veri toplama sırasında telefonunuzun sabah 10:00 ve akşam 23:00 saatleri arasında açık olması gerekmektedir. Uygulamamız sayesinde aşağıda listelenen verileri toplayabilmekteyiz:

- Telefonun donanımı (model, MAC adresi, vb.), Android İşletim Sistemi Bilgileri (ürün yazılım versiyonu, sdk, vb.)
- GPS, Bluetooth, WiFi, baz istasyonu verileri
- Yakınlık, ışık, ivmeölçer sensör verileri
- Şarj seviyesi, çalıştırılan uygulamalar, kurulu uygulamalar
- Rehber, telefon görüşmesi, mesajlaşma geçmiş özeti, tarayıcı kullanımı

Toplanan veriler içerisinde hassas bilgiler yer almamaktadır. Örneğin, telefon aramalarında konuşmanın içeriği kaydedilmemektedir. Sadece ne zaman görüşme yapıldığı bilinebilmektedir. Mesajlarda da aynı şekilde sadece ne zaman mesaj atıldığı veya alındığı

Table A.1. Participant consent form in Turkish (cont.).

kaydedilmekte fakat mesaj içerikleri kaydedilmemektedir. Telefon ile arayan veya mesaj atan kişilerin isimleri ve telefon numaraları kaydedilmemektedir. Tarayıcı aramalarında da aramanın yapıldığı zaman kaydedilmekte fakat aramada kullanılan kelimeler, ziyaret edilen sayfalar kaydedilmemektedir. Tüm telefon numaraları (rehberdeki, mesaj atılan/alınan ve telefon aramasında kullanılan) tek yönlü harmanlanmaktadır (hashing). Bu sayede örneğin her gün düzenli olarak aranılan bir kişinin telefon numarasının harmanı her gün aynı değeri vereceğinden bu davranış tarafımızdan anlaşılabilmekte fakat aranılan kişinin kim olduğu (telefon numarası, ismi, vb.) bilinememektedir.

Uygulama sürekli çalışmayıp ara ara aktif olmaktadır. Verilerin toplanma sıklığı telefon şarjınızın bir gün idare edebileceği şekilde ayarlanmıştır.

Telefonunuzun mikrofonu çevrenizdeki çeşitli sesleri (örneğin arkada çalan radyonun sesi, sokak gürültüsü veya arkadaşınızla ettiğiniz bir sohbet) ara ara kaydetmek için programlanmıştır. Her ses kaydının süresi çok kısa olduğundan (sadece 1 dakika) konuşuyorsanız, söylediklerinizin sadece çok küçük bir yüzdesi kaydedilecektir. Yine de özel yaşam ve mahremiyetinizi korumak için, uygulamamız size tüm ses kayıtlarınızı dinleme şansı verecektir. Bizimle paylaşmak istemediğiniz ses kayıtlarını silme hakkınız vardır.

Çalışmaya katılımın risk ve faydaları:
Bu araştırmaya katılmanın herhangi bir fiziksel veya psikolojik riski yoktur. Faydası ise gün içinde kendi duygu ve düşüncelerinizi takip etme ve inceleme şansına sahip olmanızdır. Önceki çalışmalarımızdan gözlemlediğimiz, katılımcıların bu deneyimi ilginç ve faydalı bir iç gözlem süreci olarak tanımladıklarıdır.

Katılım Bedeli:
Katılımınız karşılığında 150 Türk Lirası kazanacaksınız. Eğer çalışmaya geç dahil olduysanız veya erken bıraktıysanız, katılım gösterdiğiniz gün sayısına göre ödeme yapılacaktır.

Gizlilik:
Sizden toplanan her veri gizli tutulacak ve saklanacaktır. İsminiz hiç bir yerde kesinlikle geçmeyecek ve verinizle bağdaştırılmayacaktır. Verinize sadece ve sadece bu çalışmanın araştırmacıları (Dr. Bert Arnrich, Dr. Burcu Demiray Batur ve Hasan Faik Alan) ulaşabilecektir.

ANLAŞMA:

Yukarıdaki bilgileri okudum. Bu çalışmaya kendi istek ve rızamla katılmış bulunmaktayım.

Katılımcı: ____________________________________ Tarih: ________________

İmza: ____________________________________

A.2. Survey Questions

Table A.2. Thinking survey questions.

1. Just now, what were you thinking about?
Record your voice
2. How much were your thoughts focused on: (slide the bar for all three options from not at all to very much).
• The past • The future • Your current activity in the present moment
3. Have these thoughts suddenly popped up in your mind by themselves (without an attempt to think about them)?
Yes or No (Tick only one)
4. The affective content of your thoughts were:
(Slide the bar from very negative to very positive).
5.These thoughts served the following function(s) and helped you to: (Only slide the bar for those that apply, from not at all to very much):
• make a decision • achieve a goal • plan an action • solve a problem • help you move forward with your current task/activity • empathize with someone/understand someone • feel close to someone/connect with someone • understand yourself • reassure your beliefs and values • remind yourself what kind of a person you are

Table A.2. Thinking survey questions (cont.).

• reduce boredom • daydream
6. Just now, how much do you feel: (slide the bar from not at all to very much).
• Happy: • Stressed: • Awake:
7. Just before the beep, what were you doing? Please tick all that apply.
• Working/studying • In a meeting, seminar, class • Commuting, traveling • In traffic • Shopping, errands • Housework, chores • Cooking, preparing food • Admin, finances, organizing • Waiting, waiting in line • Childcare, playing with children • Petcare, playing with pets • Care or help for adults • Talking, chatting, socializing • Texting, email, social media • Browsing the internet • Watching TV/film

Table A.2. Thinking survey questions (cont.).

• Listening to music • Theater, cinema, concert • Sports, running, exercise • Computer games, smartphone games, other games • Hobbies, arts, crafts, • Reading • Washing, dressing, grooming • Walking, taking a walk • Intimacy, making love • Sleeping, resting, relaxing • Sick in bed • Meditating, religious activities • Eating, snacking • Drinking tea/coffee • Drinking alcohol • Smoking
8. To what extent were you focused/concentrated on what you were doing?
(slide the bar from not at all to very much)
9. Are you alone, or with strangers only? Or are you with your: (Please tick all that apply)
• Alone • Spouse, partner, girl/boyfriend • Children • Other family members • Colleagues, classmates

Table A.2. Thinking survey questions (cont.).

• Boss • Clients, customers • Friends • Other people you know • People I don't know
10. Where are you?
• Home • Work • In a vehicle • Restaurant/Cafe/Bar • Supermarket • Outdoor/Public Place

Table A.3. Talking survey questions.

1. How much were your talk focused on (slide the bar for all three options from not at all to very much)
• The past • The future • Your current activity in the present moment
2. This last conversation served the following function(s) and helped you to: (Only slide the bar for those that apply, from not at all to very much):
• make a decision • achieve a goal • plan an action • solve a problem • help you move forward with your current task/activity • teach/inform someone • empathize with someone/understand someone • feel close to someone/connect with someone • get to know someone • understand yourself • reassure your beliefs and values • remind yourself what kind of a person you are • reduce boredom
3. Did this conversation affect your mood? (Tick only one)
• Worse

Table A.3. Talking survey questions (cont.).

• Better • No impact
4. Just before the beep, what were you doing? Please tick all that apply.
• Talking, chatting, socializing • Working/studying • In a meeting, seminar, class • Commuting, traveling • In traffic • Shopping, errands • Housework, chores • Cooking, preparing food • Admin, finances, organizing • Waiting, waiting in line • Childcare, playing with children • Petcare, playing with pets • Care or help for adults • Texting, email, social media • Browsing the internet • Watching TV/film • Listening to music • Theater, cinema, concert • Sports, running, exercise • Computer games, Iphone games, other games • Hobbies, arts, crafts,

Table A.3. Talking survey questions (cont.).

• Reading • Washing, dressing, grooming • Walking, taking a walk • Intimacy, making love • Sleeping, resting, relaxing • Sick in bed • Meditating, religious activities • Eating, snacking • Drinking tea/coffee • Drinking alcohol • Smoking
5. To what extent were you focused/concentrated on what you were doing?
(slide the bar from not at all to very much)
6. Who were you just talking to? (Please tick all that apply)
• Spouse, partner, girl/boyfriend • Children • Other family members • Colleagues, classmates • Boss • Clients, customers • Friends • Other people you know • Stranger (s)
7. To what extent do you like the person/people you were talking to?
(Slide the bar from not at all to very much)

Table A.3. Talking survey questions (cont.).

8. To what extent do you evaluate this conversation as intentionally sought out?
(Slide the bar from not at all to very much)
9. Where are you?
• Home • Work • In a vehicle • Restaurant/Cafe/Bar • Supermarket • Outdoor/Public Place

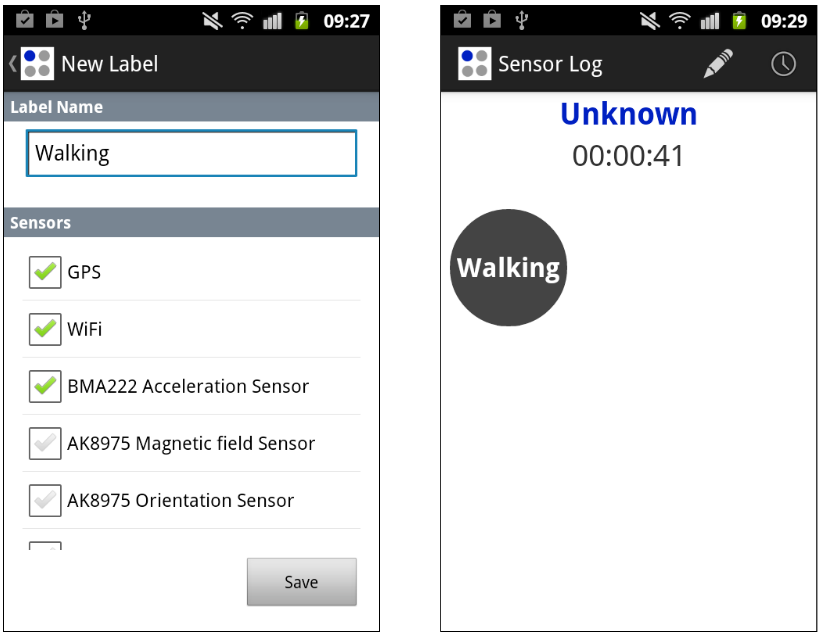
APPENDIX B: UTILITY APPLICATIONS

B.1. Sensor Log: A Mobile Data Collection and Annotation Application

The sensory data collected from smartphones need to be partially or completely annotated in order to be used in the training phase of learning algorithms. In order to ease the time consuming and burdensome annotation process, a mobile data collection and annotation application named Sensor Log is developed [59]. The application can be downloaded from the Android applications distribution platform [5] .

B.1.1. Application Usage

A new label is defined by giving it a name and choosing associated sensors from available device sensors (Figure B.1a). Defined label appears as a round button on the application home screen (Figure B.1b).

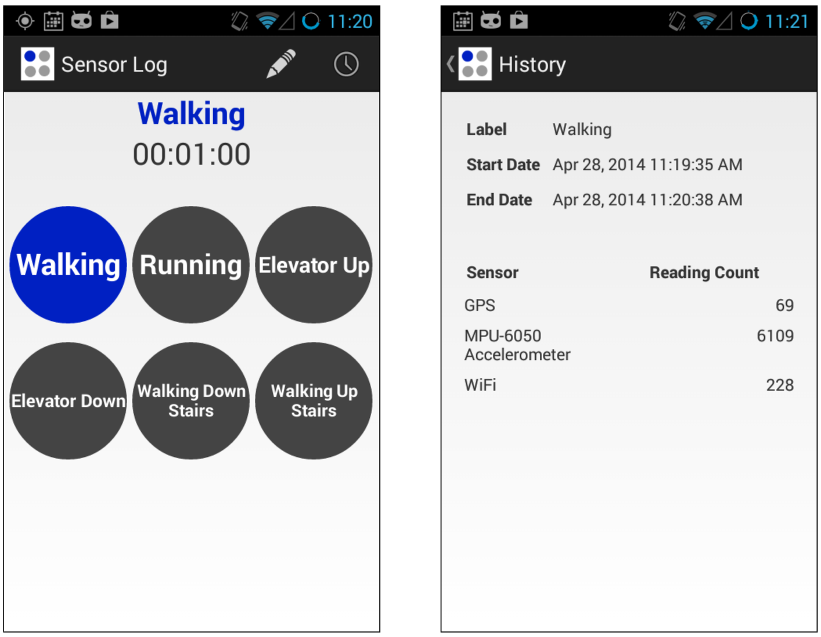


(a) (b)

Figure B.1. New label definition: (a) associating sensors with a label (b) created label is shown on the home screen as a round button.

[5]http://hfalan.com/sensorlog

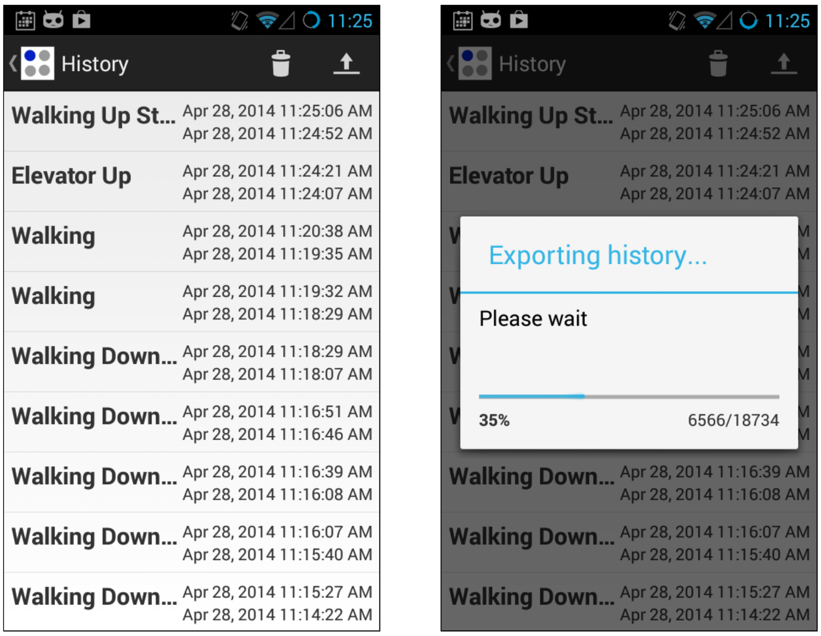
When a button on the application's home screen is tapped, all the sensors associated with the label begin collecting data and the color of the button turns to blue (Figure B.2a). Data collection is performed via a Service, an Android operating system application component, which allows to continue data collection in the background even when the application or the screen of the phone is closed. Data are requested from all sensors at the maximum sampling rate. For example, using Galaxy S3 Mini I8190 smartphone, in about 63 seconds, 6109 accelerometer data are collected which corresponds to approximately 100 Hz of sampling rate (Figure B.2b). A constant sampling rate cannot be guaranteed by the operating system due to system load fluctuations and other applications that may be requesting sensory data at the same time. Also there are sensor specific constraints for the sampling rate. For example, GPS data may not be collected at all if the phone is in an indoor place. When the active button (i.e., blue colored button) is tapped again data collection stops.



(a) (b)

Figure B.2. Data collection user interfaces in Sensor Log: (a) when a button is tapped all the associated sensor begin collecting data (b) number of data points collected from each sensor are shown on the History screen.

Each data collection session is listed on the History screen with start and end times (Figure B.3a). A single session or all of the sessions can be deleted from the list. Deletion feature allows to have a quality data set by eliminating unwanted sessions in the data. When a list item is chosen, a detail screen is shown for the chosen session (Figure B.3b). In the detail screen, sensor reading count is shown for each sensor associated with the label.



(a) (b)

Figure B.3. User interfaces for reviewing history and data export.

Collected data are stored in an SQLite database named *sensor.db* in the internal storage of the device. Files stored in the internal storage are accessible only to the application using them. In order to access the SQLite database, export button with an arrow icon is tapped on the History screen. The database is then exported to the external storage of the device where a file can be accessed by anyone. An SQLite browser software or a programming language that has an interface to communicate with an SQLite database can be used to review and analyze the collected data. It is also offered as a functionality to export the collected data in CSV file format and send the file as an email attachment.

B.1.2. Sensor Log Feasibility Study 1 - Transportation Mode Discovery

In this study Sensor Log is used for collecting and labeling GPS and Accelerometer data for discovering the transportation mode of a person carrying a smartphone. Data collection for this study is performed by a single person carrying a Samsung Galaxy S3 Mini I8190 smartphone. Running, Walking and In Vehicle labels are defined in the application. Each label is associated with GPS and accelerometer. Running and walking activities are performed in a stadium which has an oval running track (Figure B.4). In Vehicle activity is performed by traveling by bus in a city (Figure B.5).

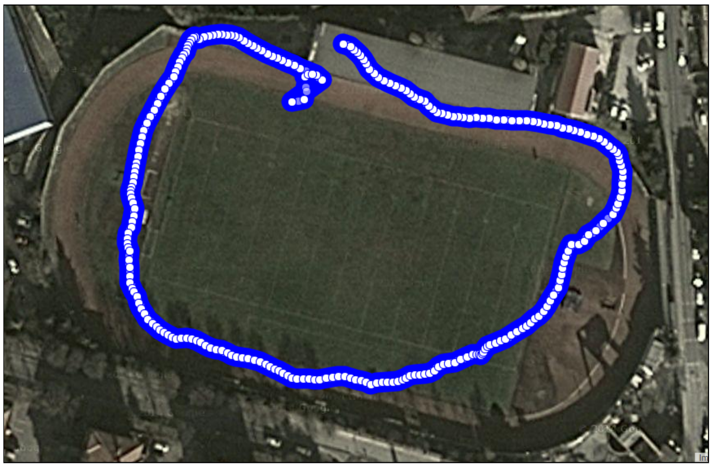


Figure B.4. GPS data points for a sample Running activity are shown on top of an oval running track.

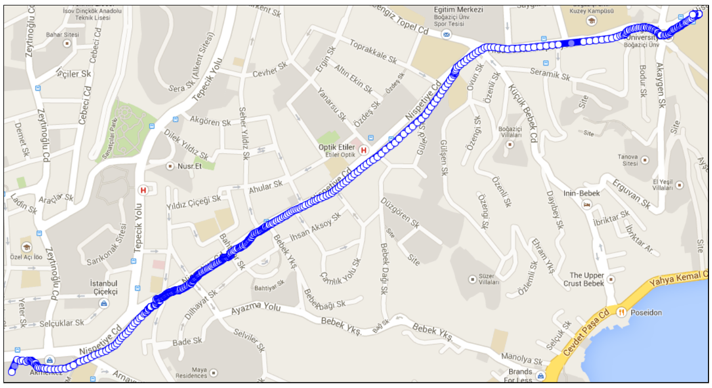


Figure B.5. GPS data points for a sample In Vehicle activity are shown.

Details of the collected data are given in Table B.1. The data are split into five seconds long windows and mean speed value is calculated for each window. If mean speed value for a window is below 2 m/s, transportation mode for this window is determined as Walking. If mean speed value is between 2 m/s and 8 m/s, transportation mode is determined as Running and if mean speed is higher than 8 m/s, the mode is determined as In Vehicle. Using this simple heuristic, transportation mode of the person is guessed for 150 windows (Table B.2).

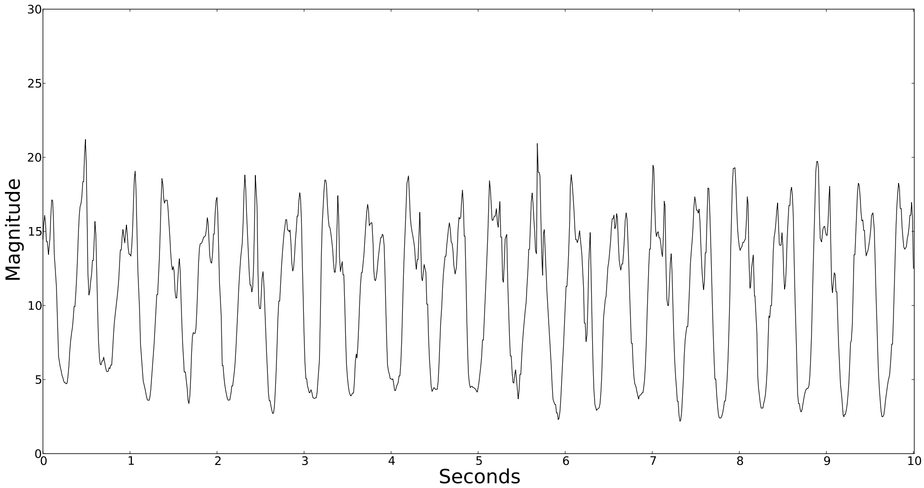
Table B.1. Details of the collected data for different transportation modes.

	Duration	**GPS**	**Accelerometer**
Walking	4 minutes 25 seconds	258	25246
Running	2 minutes 38 seconds	151	15200
In Vehicle	9 minutes 6 seconds	570	56382

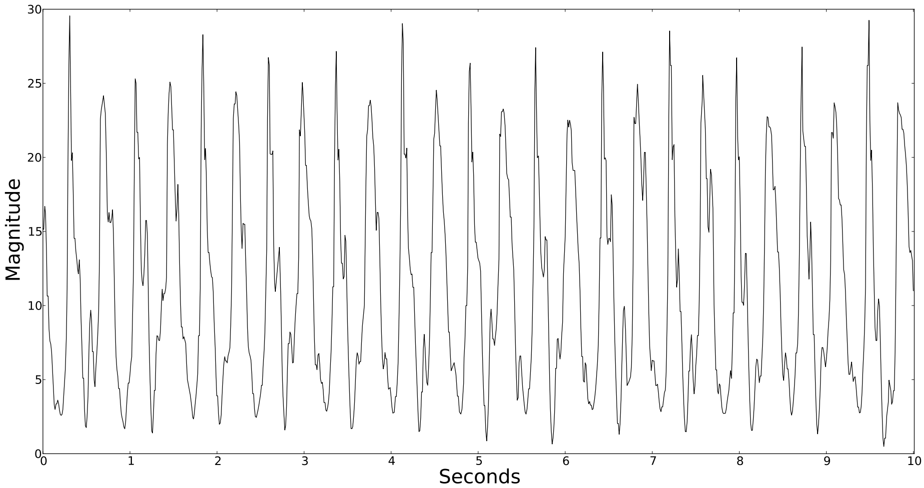
Table B.2. Confusion table of the transportation mode discovery study.

	Walking	*Running*	*In Vehicle*
Walking	50	0	0
Running	16	34	0
In Vehicle	0	6	44

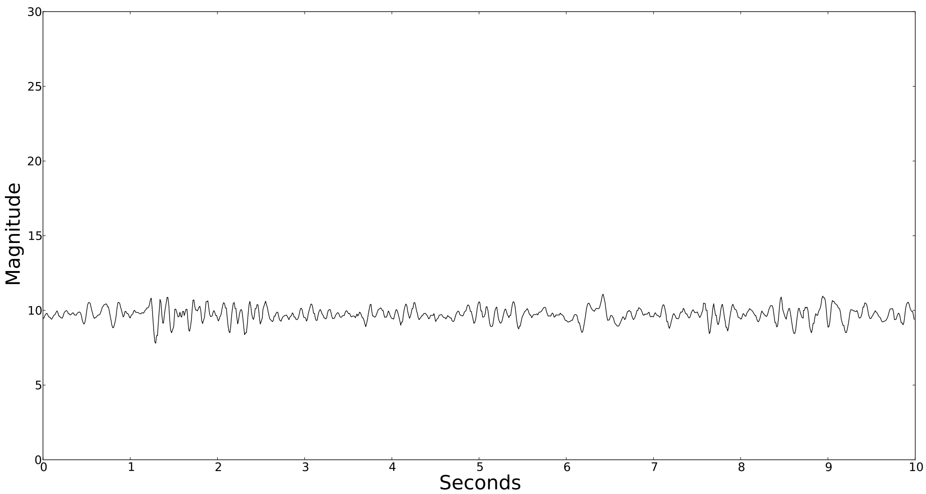
In this feasibility study, we used only the speed of a smartphone obtained from GPS to discover the transportation mode of a person. However, in Figure B.6, a distinct pattern is observed in accelerometer magnitude values for different transportation modes. By extracting various features from accelerometer magnitude values (e.g., mean, variance, max-min, etc.) and training a machine learning algorithm (e.g., Decision Tree (DT), K-nearest neighbor (KNN), Support Vector Machine (SVM), etc.) using these features we can get better results as it is shown in the literature [7].



(a)



(b)



(c)

Figure B.6. Acceleration magnitude patterns of a smartphone for three different transportation modes: (a) walking (b) running (c) in vehicle.

B.1.3. Sensor Log Feasibility Study 2 - Indoor Localization

In this study Sensor Log is used to collect and label Wi-Fi access point data for an indoor localization study. The data are collected using Samsung Galaxy W I8150 smartphone. Six different place labels namely Office Room 1, Office Room 2, Office Kitchen, Home Kitchen, Home Living Room and Supermarket are defined in Sensor Log. All of the place labels are associated with Wi-Fi. Each label is activated for about five minutes in the corresponding location. Number of Wi-Fi scans performed in each place is shown in Table B.3

Table B.3. Number of Wi-Fi scans performed for six different places.

Place Names	**Wi-Fi Scan Count**
Office Room 1 (P1)	236
Office Room 2 (P2)	72
Office Kitchen (P3)	71
Home Kitchen (P4)	71
Home Living Room (P5)	81
Supermarket (P6)	73

For each place, five of the Wi-Fi scans are used for training and 60 scans are used for testing. 111 unique BSSIDs are found in the training Wi-Fi scans. 111 bits long bit strings are generated for each place as it is shown in Figure B.7. A 1 in the bit string of a location denotes corresponding BSSID was seen in at least one of the training Wi-Fi scans of this location. These bit strings constitute the Wi-Fi fingerprints of the locations. Similarly 60 test fingerprints are generated from the test Wi-Fi scans.

Each of the 60 test Wi-Fi fingerprints are then compared with fingerprints of the locations and a test scan is determined as belonging to a place which has the most similar WiFi fingerprint to its fingerprint. Similarity is measured using Tanimoto similarity as it is defined in Figure **??**. Confusion table is shown in Table B.4.

```
Office Room 1     [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0,...,0]
Office Room 2     [1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,...,1]
Office Kitchen    [1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,...,0]
Home Kitchen      [0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1,...,0]
Home Living Room  [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,...,0]
Supermarket       [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0,...,0]
```

Figure B.7. Wi-Fi fingerprints of 6 different locations.

Table B.4. Confusion table of the indoor localization study.

	P1	P2	P3	P4	P5	P6
Office Room 1 (P1)	60	0	0	0	0	0
Office Room 2 (P2)	0	60	0	0	0	0
Office Kitchen (P3)	0	0	60	0	0	0
Home Kitchen (P4)	0	0	0	59	1	0
Home Living Room (P5)	0	0	0	6	54	0
Supermarket (P6)	0	0	0	0	0	60

The accuracy of the localization algorithm may be improved if we take into account the received signal strength values (RSS) of APs. RSS values can be assumed to come from a Gaussian distribution. During training, mean and variance of the RSS values of each AP can be calculated in each location. These values can then be used as parameters of the Gaussian functions designed for each AP. Given a test scan, we can get a score for a probable place by multiplying the values we get from Gaussian functions designed for each AP in the location. The test RSS values are the input of the Gaussian functions. The location which produces the highest score is then determined as the current place of the smartphone. We developed a utility application named Wi-Fi Indoor Localization that uses this algorithm for localization. It can be downloaded from the Android applications distribution platform [6] .

[6]http://hfalan.com/wifilocalization

REFERENCES

1. Schilit, B., N. Adams and R. Want, “Context-aware Computing Applications”, *Workshop on Mobile Computing Systems and Applications (WMCSA)*, pp. 85–90, 1994.

2. Brown, P. J., “The Stick-e Document: a Framework for Creating Context-aware Applications”, *Proceedings of the Electronic Publishing*, Vol. 8, pp. 259–272, 1996.

3. Schilit, B. N. and M. M. Theimer, “Disseminating Active Map Information to Mobile Hosts”, *Network*, Vol. 8, No. 5, pp. 22–32, 1994.

4. Dey, A. K., “Understanding and Using Context”, *Personal and Ubiquitous Computing*, Vol. 5, No. 1, pp. 4–7, 2001.

5. Samsung, *Samsung Galaxy S® 5 (Virgin Mobile), Shimmery White*, 2014, `http://www.samsung.com/us/mobile/cell-phones/SM-G900PZWAVMU`, [Accessed May 2014].

6. Kose, M., O. D. Incel and C. Ersoy, “Online Human Activity Recognition on Smart Phones”, *Workshop on Mobile Sensing: From Smartphones and Wearables to Big Data*, pp. 11–15, 2012.

7. Reddy, S., M. Mun, J. Burke, D. Estrin, M. Hansen and M. Srivastava, “Using Mobile Phones to Determine Transportation Modes”, *ACM Transactions on Sensor Networks (TOSN)*, Vol. 6, No. 2, p. 13, 2010.

8. Lu, H., W. Pan, N. D. Lane, T. Choudhury and A. T. Campbell, “SoundSense: Scalable Sound Sensing for People-centric Applications on Mobile Phones”, *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*, pp. 165–178, 2009.

9. Sellen, A. J., A. Fogg, M. Aitken, S. Hodges, C. Rother and K. Wood, "Do Lifelogging Technologies Support Memory for the Past?: an Experimental Study Using Sensecam", *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 81–90, 2007.

10. Lee, M. L. and A. K. Dey, "Lifelogging Memory Appliance for People with Episodic Memory Impairment", *Proceedings of the 10th International Conference on Ubiquitous Computing*, pp. 44–53, 2008.

11. Gartner, *Gartner Says Annual Smartphone Sales Surpassed Sales of Feature Phones for the First Time in 2013*, 2014, `http://www.gartner.com/newsroom/id/2665715`, [Accessed June 2014].

12. Google, *Android Compatibility*, 2014, `http://source.android.com/compatibility/`, [Accessed June 2014].

13. Bush, V., "As We May Think", *The Atlantic Monthly*, Vol. 176, No. 1, pp. 101–108, 1945.

14. Gemmell, J., G. Bell, R. Lueder, S. Drucker and C. Wong, "MyLifeBits: Fulfilling the Memex Vision", *Proceedings of the Tenth ACM International Conference on Multimedia*, pp. 235–238, 2002.

15. Gemmell, J., G. Bell and R. Lueder, "MyLifeBits: a Personal Database for Everything", *Communications of the ACM*, Vol. 49, No. 1, pp. 88–95, 2006.

16. Hodges, S., L. Williams, E. Berry, S. Izadi, J. Srinivasan, A. Butler, G. Smyth, N. Kapur and K. Wood, "SenseCam: A Retrospective Memory Aid", *UbiComp 2006: Ubiquitous Computing*, pp. 177–193, 2006.

17. "SenseCam", `http://research.microsoft.com/en-us/um/cambridge/projects/sensecam/`, [Accessed May 2014].

18. Aizawa, K., "Digitizing Personal Experiences: Capture and Retrieval of Life Log",

In Proceedings of the 11th International Multimedia Modelling Conference, pp. 10–15, 2005.

19. Abe, M., D. Fujioka and H. Handa, “A Life Log Collecting System Supported by Smartphone to Model Higher-level Human Behaviors”, *Sixth International Conference on Complex, Intelligent and Software Intensive Systems*, pp. 665–670, 2012.

20. Vemuri, S., C. Schmandt, W. Bender, S. Tellex and B. Lassey, “An Audio-based Personal Memory Aid”, *UbiComp 2004: Ubiquitous Computing*, pp. 400–417, 2004.

21. Frank, J., S. Mannor and D. Precup, “Generating Storylines from Sensor Data”, *Pervasive and Mobile Computing*, Vol. 9, No. 6, pp. 838–847, 2013.

22. Teh, Y. W., M. I. Jordan, M. J. Beal and D. M. Blei, “Hierarchical Dirichlet Processes”, *Journal of the American Statistical Association*, Vol. 101, No. 476, 2006.

23. Chennuru, S., P.-W. Chen, J. Zhu and J. Y. Zhang, “Mobile Lifelogger–Recording, Indexing, and Understanding a Mobile User’s Life”, *Mobile Computing, Applications, and Services*, pp. 263–281, 2012.

24. Blum, M., A. Pentland and G. Troster, “Insense: Interest-based Life Logging”, *IEEE MultiMedia*, Vol. 13, No. 4, pp. 40–48, 2006.

25. Marmasse, N. and C. Schmandt, “Location-Aware Information Delivery withComMotion”, *Handheld and Ubiquitous Computing*, pp. 157–171, Springer, 2000.

26. Ashbrook, D. and T. Starner, “Using GPS to Learn Significant Locations and Predict Movement Across Multiple Users”, *Personal and Ubiquitous Computing*, Vol. 7, No. 5, pp. 275–286, 2003.

27. Montoliu, R., J. Blom and D. Gatica-Perez, “Discovering Places of Interest in Everyday Life from Smartphone Data”, *Multimedia Tools and Applications*, Vol. 62, No. 1, pp. 179–207, 2013.

28. Ester, M., H.-P. Kriegel, J. Sander and X. Xu, "A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.", *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*, Vol. 96, pp. 226–231, 1996.

29. Zheng, V. W., Y. Zheng, X. Xie and Q. Yang, "Collaborative Location and Activity Recommendations with Gps History Data", *Proceedings of the 19th International Conference on World Wide Web*, pp. 1029–1038, 2010.

30. Kang, J. H., W. Welbourne, B. Stewart and G. Borriello, "Extracting Places from Traces of Locations", *Proceedings of the 2nd ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots*, pp. 110–118, 2004.

31. Schilit, B. N., A. LaMarca, G. Borriello, W. G. Griswold, D. McDonald, E. Lazowska, A. Balachandran, J. Hong and V. Iverson, "Challenge: Ubiquitous Location-aware Computing and the Place Lab Initiative", *Proceedings of the 1st ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots*, pp. 29–35, 2003.

32. Ye, Y., Y. Zheng, Y. Chen, J. Feng and X. Xie, "Mining Individual Life Pattern Based on Location History", *Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*, pp. 1–10, 2009.

33. LaMarca, A., Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter *et al.*, "Place Lab: Device Positioning Using Radio Beacons in the Wild", *Pervasive Computing*, pp. 116–133, 2005.

34. Laasonen, K., M. Raento and H. Toivonen, "Adaptive On-device Location Recognition", *Pervasive Computing*, pp. 287–304, 2004.

35. Hightower, J., S. Consolvo, A. LaMarca, I. Smith and J. Hughes, "Learning and Recognizing the Places We Go", *UbiComp 2005: Ubiquitous Computing*, pp. 159–176, 2005.

36. Kim, D. H., J. Hightower, R. Govindan and D. Estrin, "Discovering Semantically Meaningful Places from Pervasive RF-beacons", *Proceedings of the 11th International Conference on Ubiquitous Computing*, pp. 21–30, 2009.

37. Kim, D. H., K. Han and D. Estrin, "Employing User Feedback for Semantic Location Services", *Proceedings of the 13th International Conference on Ubiquitous Computing*, pp. 217–226, 2011.

38. Froehlich, J., M. Y. Chen, S. Consolvo, B. Harrison and J. A. Landay, "MyExperience: a System for in situ Tracing and Capturing of User Feedback on Mobile Phones", *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*, pp. 57–70, 2007.

39. Aharony, N., W. Pan, C. Ip, I. Khayal and A. Pentland, "Social fMRI: Investigating and Shaping Social Mechanisms in the Real World", *Pervasive and Mobile Computing*, Vol. 7, No. 6, pp. 643–659, 2011.

40. Miluzzo, E., N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng and A. T. Campbell, "Sensing Meets Mobile Social Networks: the Design, Implementation and Evaluation of the Cenceme Application", *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, pp. 337–350, 2008.

41. Wu, P., J. Zhu and J. Y. Zhang, "Mobisens: a Versatile Mobile Sensing Platform for Real-world Applications", *Mobile Networks and Applications*, Vol. 18, No. 1, pp. 60–80, 2013.

42. Lane, N. D., M. Mohammod, M. Lin, X. Yang, H. Lu, S. Ali, A. Doryab, E. Berke, T. Choudhury and A. Campbell, "Bewell: A Smartphone Application to Monitor, Model and Promote Wellbeing", *5th International ICST Conference on Pervasive Computing Technologies for Healthcare*, pp. 23–26, 2011.

43. de Montjoye, Y.-A., J. Quoidbach, F. Robic and A. S. Pentland, "Predicting Per-

sonality Using Novel Mobile Phone-based Metrics", *Social Computing, Behavioral-Cultural Modeling and Prediction*, pp. 48–55, 2013.

44. LiKamWa, R., Y. Liu, N. D. Lane and L. Zhong, "Can Your Smartphone Infer Your Mood?", *Proc. PhoneSense Workshop*, 2011.

45. Shepard, C., A. Rahmati, C. Tossell, L. Zhong and P. Kortum, "LiveLab: Measuring Wireless Networks and Smartphone Users in the Field", *ACM SIGMETRICS Performance Evaluation Review*, Vol. 38, No. 3, pp. 15–20, 2011.

46. Moturu, S. T., I. Khayal, N. Aharony, W. Pan and A. S. Pentland, "Using Social Sensing to Understand the Links Between Sleep, Mood, and Sociability", *2011 IEEE Third International Conference on Social Computing (socialcom)*, pp. 208–214, 2011.

47. Wang, Y., J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari and N. Sadeh, "A Framework of Energy Efficient Mobile Sensing for Automatic User State Recognition", *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*, pp. 179–192, 2009.

48. Do, T. M. T., J. Blom and D. Gatica-Perez, "Smartphone Usage in the Wild: a Large-scale Analysis of Applications and Context", *Proceedings of the 13th International Conference on Multimodal Interfaces*, pp. 353–360, 2011.

49. Falaki, H., R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan and D. Estrin, "Diversity in Smartphone Usage", *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, pp. 179–194, 2010.

50. Oliver, E., "The Challenges in Large-scale Smartphone User Studies", *Proceedings of the 2nd ACM International Workshop on Hot Topics in Planet-scale Measurement*, p. 5, 2010.

51. Lu, H., J. Yang, Z. Liu, N. D. Lane, T. Choudhury and A. T. Campbell, "The

Jigsaw Continuous Sensing Engine for Mobile Phone Applications", *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pp. 71–84, 2010.

52. Larson, R. and M. Csikszentmihalyi, "The Experience Sampling Method.", *New Directions for Methodology of Social & Behavioral Science*, 1983.

53. Google, *Sensors Overview*, 2014, `http://developer.android.com/guide/topics/sensors/sensors_overview.html`, [Accessed May 2014].

54. Google, *AlarmManager — Android Developers*, 2014, `http://developer.android.com/reference/android/app/AlarmManager.html`, [Accessed May 2014].

55. Rogers, D. J. and T. T. Tanimoto, "A Computer Program for Classifying Plants", *Science*, Vol. 132, No. 3434, pp. 1115–1118, 1960.

56. Quercia, D., N. Lathia, F. Calabrese, G. Di Lorenzo and J. Crowcroft, "Recommending Social Events from Mobile Phone Location Data", *10th International Conference on Data Mining (ICDM)*, pp. 971–976, 2010.

57. WorldWeatherOnline®, *Global Weather Forecast and Weather API Provider*, 2014, `http://www.worldweatheronline.com/`, [Accessed June 2014].

58. WorldWeatherOnline®, *Weather Condition Codes*, 2014, `http://www.worldweatheronline.com/feed/wwoConditionCodes.txt`, [Accessed June 2014].

59. Alan, H. F., B. Arnrich, C. Ersoy and B. Cinaz, "Sensor Log: a Mobile Data Collection and Annotation Application", *Signal Processing and Communications Applications Conference (SIU), 2014 22nd*, pp. 1375–1378, 2014.