ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Data Sharing in Participatory Social Sensing

Master Thesis

Ramapriya Sridharan

September 3, 2016

Advisors: Prof. Dr. Dirk Helbing, Dr. Pournaras Evangelos
Department of Computational Social Sciences , ETH Zürich

Contents

Abstract

Data from citizens needs to be collected and analyzed to create or improve current services in society. Data collected from them, in general, reveals information about their behavior and choices. In addition, it can also reveal sensitive information, that they might not be comfortable with. To preserve the privacy of citizens is where data privacy comes into play. There are various methods to maintain data privacy and different levels of privacy to maintain. The higher the privacy level, the more concealed the data is. Given the choice, citizens would generally choose the highest privacy level. At times, less concealed data is needed while solving problems that need data with less errors. To help citizens reduce the level of privacy of the data when needed, different kinds of incentives can be used, such as monetary incentives. From a fixed budget on the demand side, rewards(incentives) are handed out to citizens to incite them to give less privatized data, yet maintaining a minimum level of privacy. The goal of the Thesis is to understand the social dynamics of privacy and information sharing. Existing data can be used or data can be collected for the purpose of the analysis.

Chapter 1

Introduction

Chapter 2

Related Work

Chapter 3

Computational Model

3.1 Introduction

The aim is to create a computational model that is able to collect useful information about the influence of monetary incentives on mobile data sharing. (Quote some studies that have done similar studies with no data incentives). The users are first asked some preliminary questions to form a profile about them. The model proceeds to use the user profiles formed to assign each sensor data request with a maximum achievable credit. The model attempts identifies the data requests where users might not be inclined to share mobile sensor data willingly. These data requests are assigned higher maximum obtainable costs. Similarly, the data requests where the users would want to share more mobile sensor data is assigned a lower maximum obtainable cost. This permits us to see whether incentives do indeed make a difference in mobile sensor data sharing. The model aims to identify the amount of data each user would share for a data request and assign maximum obtainable costs accordingly.

3.2 Model Intricacies

The sections below explain the various building blocks of the computational model. The Figure 3.1 provides an overview of the flow of the model.

Table 3.1: Abbreviations

Term	Short-Form	Definition		
Features	Features	Features can be the Sensor	Data Request	Request

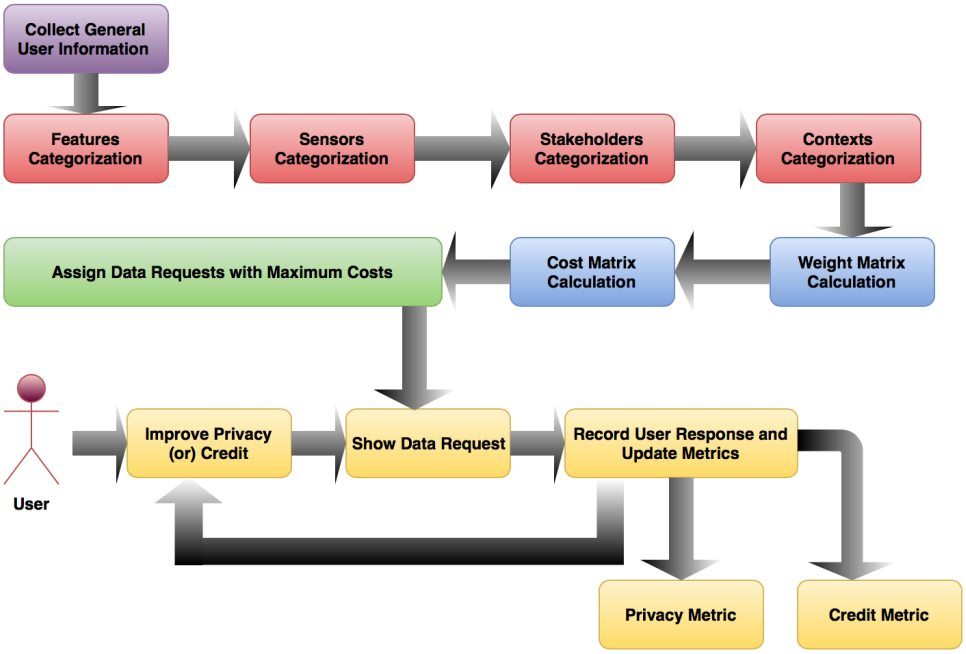


Figure 3.1: Computational Model Flow Chart

3.2.1 Collecting User Information

To begin with the model, each user is asked to enter various non-intrusive personal information. The information collected can consist of but is not limited to :

- Gender
- Year of birth
- Country
- Education Level
- Occupation
- Frequency of mobile phone use per day
- List of different mobile applications present on the users phones

Collecting this information is crucial to the data analysis that is conducted in the later stages.

3.2.2 Categorization of the Features

After the users personal information is collected, users are asked to place the Features in categories according to how privacy intrusive they are to the

user. A Feature can be one of the following:

- Sensors : Sensors consist of the sensors in the mobile phone which users can trade in a data request
- Stakeholders : Stakeholders consist of any entity that can request the user for mobile sensor data
- Contexts : Contexts consist of the purpose for which a Stakeholder would like to obtain the user's mobile sensor data

Features are the three dimensions that form a unique data request. A data request is defined as a Stakeholder asking users to share their mobile Sensor data for a particular Context.

As mentioned before, users are asked to categorize the Features into one of the five categories:

1. Very low privacy intrusion
2. Low privacy intrusion
3. Medium privacy intrusion
4. High privacy intrusion
5. Very high privacy intrusion

Categories are linearly scaled and equally spaced. As indicated by the numbers on the left of the categories, these range from one to five and users can place each of the Features in a category according to their perceived intrusion level. Category one represents that the Feature does not contribute a lot to the data sharing decision. Similarly, category five represents that the Feature contributes a lot to the user's data sharing decision. Similarly, category five represents that users are reluctant to give away their sensor data for this feature. More than one feature can be placed in the same category, which makes it a more powerful tool than the ranking mechanism.

Let the variablecat represent the number of categories, which here is five. Additionally, let the category assigned to the Sensors be represented by the variable se, the category assigned to the Stakeholders be represented by the variable st and the category assigned to the Contexts be represented by the variable co.

Once users have categorized the Sensors, Stakeholders and the Contexts into the respective categories reflecting the importance of each of the features in the data sharing decision, each feature is assigned a weight. Let the respective weights of Sensors, Stakeholders and Contexts be represented by the variables, w_{se}, w_{dc} and w_{co} are calculated as follows :

$$w_{se} = \frac{se}{se + st + co} \tag{3.1}$$

$$w_{st} = \frac{st}{se + st + co} \tag{3.2}$$

$$w_{co} = \frac{co}{se + st + co} \tag{3.3}$$

3.2.3 Categorization of the Sub-Features

Once the features have been categorized and their weights calculated as above, sub-features are to be categorized. A sub-feature is defined as one type of a feature. In other words, sub-features are the different types of features that appear during data request to the user. The following are examples of sub-features for each feature :

- Sensors :
  - Accelerometer
  - Battery
  - Gyroscope
- Stakeholders :
  - Corporation
  - Government
  - Educational Institution
- Contexts :
  - Education
  - Navigation
  - Gaming

Each of the above are different kinds or sub-features of the respective Features. For each of the available features, the respective sub-features need to be in turn categorized in a similar fashion to section 3.2.2. The categories are the same as mentioned in the previous section. Let num_{sf} be the number of sub-features each feature has.

As in the first category indicates that users find the sub-feature would not hinder the data sharing decision. This means that the user would not be worried trading data for a data request involving this sub-feature. The last category indicates that users find this sub-feature would hinder the data sharing decision . This means that users would be reluctant of giving data

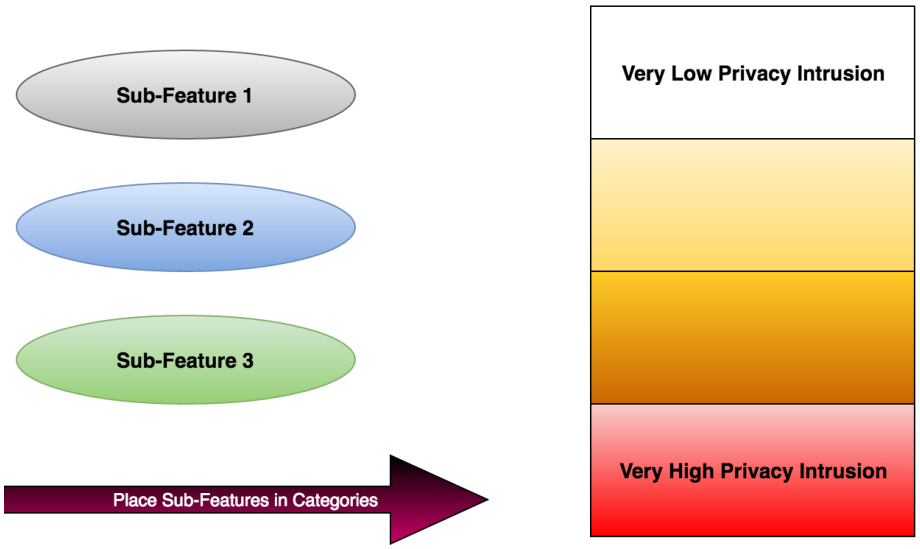


Figure 3.2: Categorizing Sub-Features according to the perceived Intrusion Level

for a data request involving this sub-feature.As seen in the conceptual diagram is shown in figure 3.2, users place each of the sub-features available for every feature in the given categories.

Let every sub-feature be represented by a unique identifier within its feature. For example, in the list of sub features provided above, accelerometer is the first sub-feature of sensors, corporation is the first sub-feature of stakeholders and education is the first sub-feature of contexts. For each of the sub-features of Sensors, categories they are placed in by users is represented by se_i and i is the identifier of the sub-feature. Similarly, categories assigned to sub features of features Stakeholders and Contexts respectively are represented by st_j and co_k, where j and k are the identifiers of the sub-features categorized.

3.2.4 Weight Matrix Calculation

Each data request to users consists of the three above mentioned features in them. Each of the features each have num_{sf} sub-features that can appear in turns in a data request. So the total number of data requests are :

$$num_{dr} = num_{sf} * num_{sf} * num_{sf} \tag{3.4}$$

Let WM be a matrix with three dimensions $num_{sf} x num_{sf} x num_{sf}$. We call this the weight matrix. Each cell of WM, that is $WM_{i,j,k}$ represents a data

request which involves the Sensors sub-feature with identifier i, Stakeholders sub-feature with identifier j, and the Contexts sub-feature with identifier k. That is, each cell of WM represents the weight of a data request to the users. The aim of the weight matrix is to use the information collected from the user categorizations, to assign weights to each data requests. Intuitively, the process examines the data requests where the user is least likely to trade data and assigns higher weights to those data requests. This process can be seen in section 3.3 with examples. As mentioned before, each cell of the matrix WM represents the weight of a data request with a unique Sensors sub-feature i, Stakeholders sub-feature j and Contexts sub-feature k. To calculate the weight of a data request :

$$WM_{i,j,k} = (se * se_i) + (st * st_j) + (co * co_k) \tag{3.5}$$

Applying this formula for every possible values of i, j and k gives the weight matrix WM.

3.2.5 Cost Matrix Calculation

Once WM has been calculated, it can give an idea of the weight each data request receives. The aim is now to assign a maximum obtainable cost to each data request. This cost is the maximum credit users can receive for a particular data request. Let CM be the cost matrix with the three dimensions $num_{sf} x num_{sf} x num_{sf}$. Let it be assumed to have a budget of b for a day, where b can be in an actual currency or any sorts of virtual credits. In this literature the budget will be referred to with the unit credits. Each cell of the cost matrix will represent the amount of credits allocated for a particular data request for one day. To begin with, we calculate the sum of all the cells of the weight matrix WM:

$$sum_{WM} = \sum_{i=1}^{num_{sf}} \sum_{j=1}^{num_{sf}} \sum_{k=1}^{num_{sf}} wm_{i,j,k} \tag{3.6}$$

where the function sum_{WM} gives the sum of a matrix, in this case the weight matrix. Let $CM_{i,j,k}$ represent the credit allocated for the data request which involves the Sensor's sub-feature with identifier i, Stakeholder's sub-feature with identifier j, and the Context's sub-feature with identifier k. To calculate one cell of the cost matrix :

$$CM_{i,j,k} = \frac{WM_{i,j,k} * b}{sum_{WM}} \tag{3.7}$$

Repeating the above for every cell of CM, the entire cost matrix can be calculated. Now, all the maximum obtainable costs have been allocated per day for every data request.

3.2.6 Cost and Privacy Metrics

Every data request now has an associated cost. This is the maximum cost that a user can obtain for that data request. The Cost metric is the total amount of credits the user has obtained for one day. Similarly, the Privacy metric is the amount of privacy percentage the user has maintained. That is, it intuitively quantifies the amount of data the user has refused to share hence implying privacy. The Cost and Privacy are inversely proportional to each other, in the sense that when the Cost goes up and Privacy goes down and vice versa. For each data request, the user can choose how much data is to be shared, from the maximum amount of data to no data at all. Each option corresponds to a summarization level explained in detail in section 3.2.8. The cost assignment to each option is linearly scaled according to the cost assigned to each data request. Let us assume there are options for a data request ranging from 1 to m (numeric options), where 1 corresponds to where the user gives all the data requested and m to where the user chooses not give any data at all. Therefore there are a total of m options for a data request. While assigning costs there are two scenarios:

- Assigning option costs without a participation cost.
- Assigning option costs inclusive of a participation cost.

Let us examine the first scenario. Let us assume that we are calculating the option costs for data request with Sensors sub-feature i, stakeholders sub-feature j and contexts sub-feature k. Let us calculate the assigned cost for option number h of this data request:

$$cost_h = \frac{CM_{i,j,k} * (m - h)}{m - 1} \tag{3.8}$$

Applying this formula by replacing h by the options from 1 to m gives the cost the user receives for each option. Similarly, if you would like to assign a participation cost to each option, it would mean that even tough the user does not share data, they still receive some money for answering the data request. This concept can be implemented to ensure user participation. (Quote some paper with participation of users in PSS). Let x be a fraction of the total budget B that is dedicated for user participation. Using a geometric progression with $a = 1$ and $r = \sqrt[(m-1)]{x}$, we can calculate the fraction of the cost $frac_h$ an option numbered h gets:

$$frac_h = a * r^{h-1} \tag{3.9}$$

Now that we know the fraction of the cost option f can be assigned, to get the cost $cost_h$ of option h for the data request with Sensors sub-feature i, stakeholders sub-feature j and contexts sub-feature k :

$$cost_h = frac_h * CM_{i,j,k} \tag{3.10}$$

This assigns costs to each option, taking into consideration a participation cost that the user gets even if data is not shared for that data request.

Privacy percentage pri_h is linearly scaled between the first to the mth option between 0 and 100 as follows:

$$pri_h = \frac{(h-1) * 100}{m-1} \tag{3.11}$$

The total cost and privacy is the arithmetic average of all the costs and privacy obtained from every answered data request. If a data request is left unanswered, maximum privacy and minimum cost is assumed.

3.2.7 Improving the Metrics

Before the user answers a question, it is useful to know what the user interest lies in. Would the user like to improve the privacy metric, or would the user would like to increase the credit revenue. In addition, if we know what the user is looking to improve, we can retrieve the question that can improve the that particular metric the most. For example if the user wishes to improve his privacy further, we look at the questions where the user has given the most amount of data. We then put forth this question to answer, which indicating all the options that can improve the privacy. Similarly, if the user chooses to obtain more credit, the question where the user has given least amount of data is retrieved. Options that can improve the user credit are also indicated.

3.2.8 Summarization of Collected Data

As mentioned before, each data request can have options m number of options the user can choose from. These options range from 1, which indicates that the user would like to give all his data, to option number m, which indicates when the user does not want to give any data to this data request. Even tough all data is encrypted these days, it is still not enough as encryptions might be cracked. Summarization is a privacy algorithms that aggregates data to provide less information than in its original form. The higher the summarization level gives less data than than in its original form. The lower the summarization level gives data closer to its original form. In this model, data is collected for a period of 24 hours every y seconds for every data

request.If the data is summarized, according to the option chosen, the data is collected either every y seconds or lesser.

Data is collected for the whole day, and at the end of the day according to the option chosen by the user, it is summarized. Summarization can be linearly assigned to each option starting with the highest privacy corresponding to highest summarization level , that is no data sharing to the lowest summarization level, that is no summarization at all. An example of assigning the summatization level $summ_h$ for option h can be the following :

$$summ_h = y * h \text{where} h \neq m \tag{3.12}$$

This gives the frequency of sensor data collection for every option of a data request.

3.3 Analysis of the Model

In this section, we take a scenario of the computational model and show how exactly the model works. In particular, the focus is on how the model varies the weights to questions according to the user input.

3.3.1 Setup

the sensors, stakeholders, and contexts and other special parameters such as number of options and all To explain the model using examples, we take into consideration the following sub-features for each feature:

1. Sensors
    a) Accelerometer -1
    b) Noise -2
    c) Location -3
2. Stakeholders
    a) Corporation -1
    b) Government -2
    c) Educational Institution -3
3. Contexts
    a) Navigation -1
    b) Environment -2

c) Social Media -3

The numbers indicated next to the sub-features is the sub-feature identifier. This uniquely identifies a sub-feature within a feature category. Each user will receive an amount of

$$count(Sensors) * count(Stakeholders) * count(Contexts) = 27$$

data requests in total. Each data request has five privacy options ranging from one to five. the option one indicates the users would like to to trade all their data, and option five indicates the users refuse to share data their for this data request. Additionally, it is assumed that the core phase has a Budget $B = 100$ per day. The input to the model are the user choices during the categorization of the features and sub-features.

3.3.2 Results

In this section, three user scenarios will be introduced and explained in order to explore the properties of the weight and cost matrices. First, we will begin by introducing the way the user has categorized the features and sub-features. This will be followed by an explanation of the generated matrices. To make reference easier to the graphs, instead of sub-feature names, numeric identifiers are used. For example, accelerometer is th Sensor's sub-feature 1. Similarly, Navigation is Context's sub-feature 1. The tuple (a,b,c) represents a data request with:

1. a - Sensor's sub-feature a
2. b - Stakeholder's sub-feature b
3. c - Context's sub-feature c

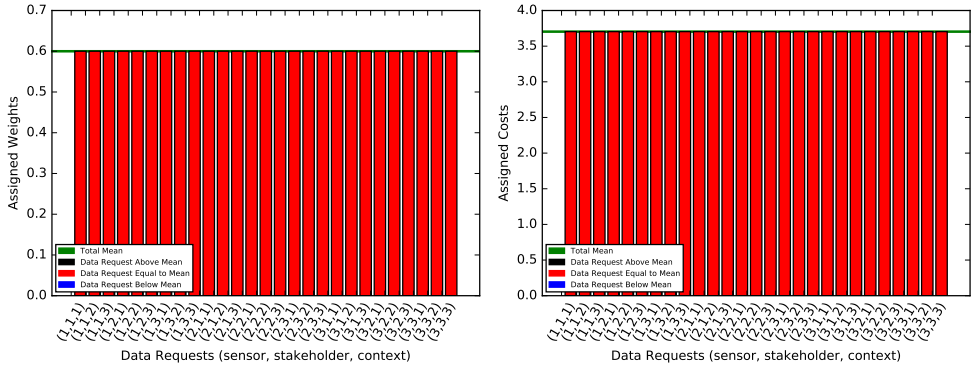
where a,b and c are all numbers from one to three.

Scenario One

In scenario 1.1, the users choose categories for the Features and sub-features as shown in the table 3.2. As it can be seen in the table, each Feature receive category 1, and all their sub-features are categorized as 3. In short, all the features have the same categorization and their respective sub-features all have the same categorization as well. From this input, the formulation of the weight matrix can be seen in figure 3.3a, and the cost matrix can be seen in figure 3.3b. As we expected, for each data request indicated as a tuple of (sensors, stakeholders, contexts) in the x-axis of figures 3.3 have identical weights and costs. This is due to the fact that the user finds all the Features and sub-features equally intrusive so all the data requests are weighted equally.

Table 3.2: Categorization for Scenario 1.1

Feature	Sub-Feature ID = 1	Sub-Feature ID = 2	Sub-Feature ID = 3
Sensors	Accelerometer	Noise	Location
1	3	3	3
Stakeholders	Corporation	Government	Educational Institution
1	3	3	3
Contexts	Navigation	Environment	Social Media
1	3	3	3



(a) Values of the Weight Matrix

(b) Values of the Cost Matrix

Figure 3.3: Examining Scenario 1.1

The theory that all equally intrusive Features and sub-features should have data requests with equal weights and costs forms the basis of the computational model. Hence, it becomes essential to view another similar scenario to confirm that this indeed works. The table 3.3 is the user input to the next scenario 1.2. Similar to scenario 1.1 but with different inputs, the Features and sub-features categorized are viewed to all be equally intrusive by the user. Hence as shown in figures 3.4a and 3.4b, data requests are again weighted equally.

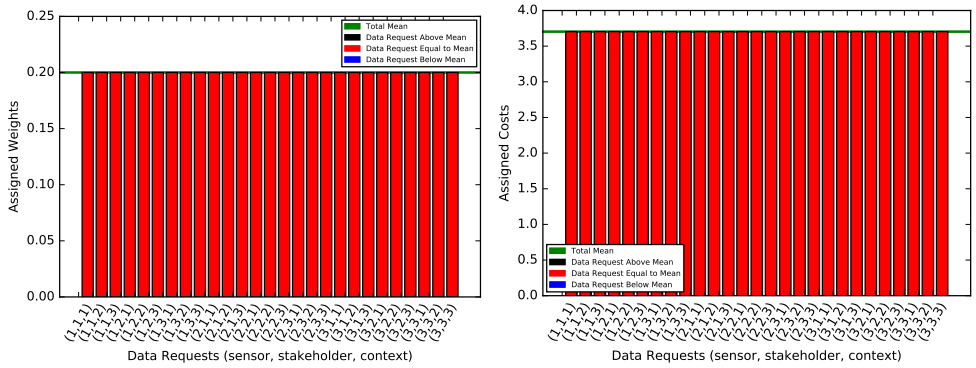
We can conclude that if the user perceives the feature and respective sub-features in an equally intrusive way, then all the data requests will have the same costs assigned.

Scenario 3

Table 3.4 indicates the user input to scenario 3. As it can be seen, all Features have equal categories, and all sub-features have the categories of 3 with an exception the Sensor's sub-features. The Sensor's sub-features with

Table 3.3: Categorization for Scenario 1.2

Feature	Sub-Feature ID = 1	Sub-Feature ID = 2	Sub-Feature ID = 3
Sensors	Accelerometer	Noise	Location
4	1	1	1
Stakeholders	Corporation	Government	Educational Institution
4	1	1	1
Contexts	Navigation	Environment	Social Media
4	1	1	1



(a) Values of the Weight Matrix

(b) Values of the Cost Matrix

Figure 3.4: Examining Scenario 1.2

identifiers 1,2 and 3 have respectively categories 1,3 and 5. This means that requests requests with Sensor's sub-feature 1 will have the lesser weight in comparison to the other Senso'r sub-features. Similarly, the data requests with Sensor's sub-feature 2 will have a higher weightage than Sensor's sub-feature 1, but lesser than Sensor's sub-feature 3. Lastly, data requests with Sensor's sub-feature 3 will have a higher weight compared to the others, due to its category being 5. The weight and cost matrices can be seen in figures 3.5a and 3.5b respectively.

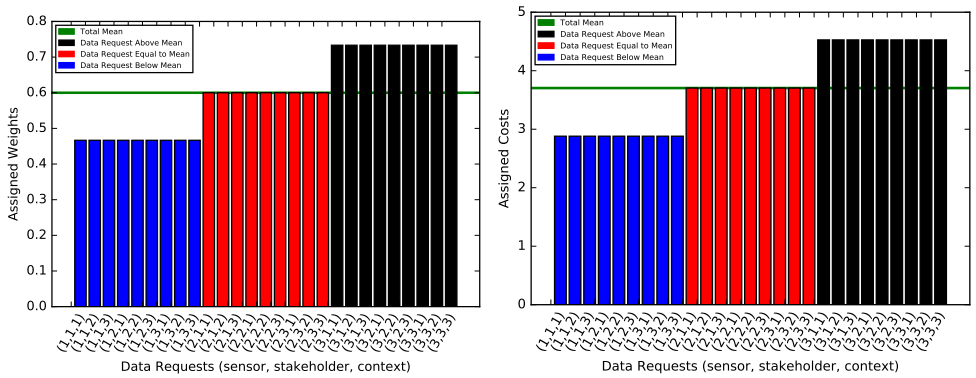
From the above input and graphs, we can conclude that the model assigns a higher weight to data requests with sub-features that the user finds more intrusive compared to the others.

Scenario 4

An attempt is made to vary the feature and sub-feature categories at once, to show how varying their values together affects the assignments of the weight matrix. Table 3.5 is the user input to the scenario 4. All the Features

Table 3.4: Categorization for Scenario 3

Feature	Sub-Feature ID = 1	Sub-Feature ID = 2	Sub-Feature ID = 3
Sensors	Accelerometer	Noise	Location
3	1	3	5
Stakeholders	Corporation	Government	Educational Institution
3	3	3	3
Contexts	Navigation	Environment	Social Media
3	3	3	3



(a) Values of the Weight Matrix

(b) Values of the Cost Matrix

Figure 3.5: Examining Scenario 3

have different categories assigned from 3 to 5. Additionally, the sub-feature 1 of each feature has a category of 5, higher than the others which are all categorized as 1. The weight and cost matrices generated for this scenario can be seen in figures 3.6a and 3.6b respectively.

As it is observed for both figures, the data request with the highest weight is the one with tuple (1,1,1). This tuple indicates that the data request involves all sub-features 1 of each feature. It happens because all of the sub-features 1 are assigned a category of 5. The feature Sensors feature and its sub-feature 1 are categorized as 5, so all the data requests with tuple (1,*,*), where * is all the other possible sub-features from other features, are all above average as seen in figures 3.6, irrespective of the categories of the other Feature's sub-features. This shows that assigning a higher category to a feature can lead to higher data request costs. The green horizontal line in the graph indicates the mean value of the weights and costs. In general due to sub-features categorized as 5, those data request receive a higher weight and cost. In some cases, the data requests still receive a lower weight such

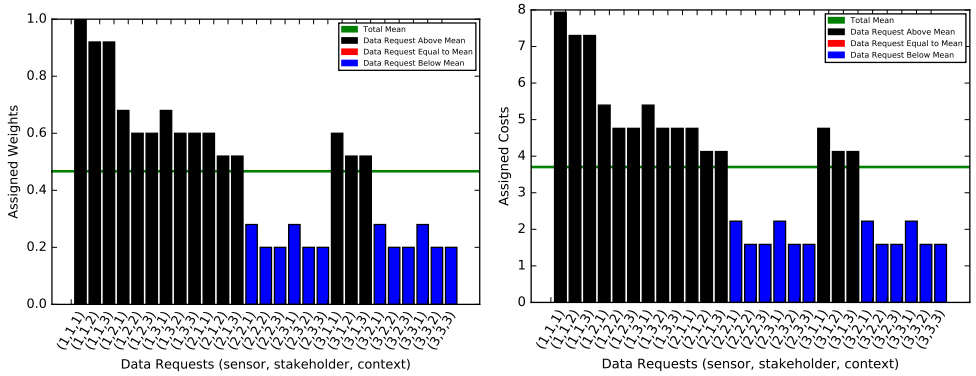
as tuple (2,2,1), (2,3,1),(3,2,1) and (3,3,1) even tough Context feature's sub-feature 1 has a category of 5. This is due to the fact that Sensor's feature and Stakeholders feature have a higher categories of 5 and 4 respectively than the context feature. Since their sub-features are assigned a lower privacy intrusion category than the context's sub-features, the weight of the data requests is lower. This shows that even tough a sub-feature may be regarded as very intrusive, it's weight increasing changing ability still depends on the category of its feature.

Additionally, it can be noted that data requests with at least two sub-features 1 are all above average. We can witness the property of the model, which puts more emphasis on the perception of the Features than the sub-features themselves. As seen in the figure, all the features with higher intrusion categorizations have weights and costs that are well above average.

We can conclude that the model assigns weights to data requests, by putting more emphasis on the feature's weights. A feature with high category has the ability to assign higher costs with a highly categorized sub-feature. It also has the ability to lower the weight with a sub-feature lowly categorized. Features with lower categories contribute lesser to the weight assignments, irrespective of their sub-feature categories.

Table 3.5: Categorization for Scenario 4

Feature	Sub-Feature ID = 1	Sub-Feature ID = 2	Sub-Feature ID = 3
Sensors	Accelerometer	Noise	Location
5	5	1	1
Stakeholders	Corporation	Government	Educational Institution
4	5	1	1
Contexts	Navigation	Environment	Social Media
3	5	1	1



(a) Values of the Weight Matrix (b) Values of the Cost Matrix

Figure 3.6: Examining Scenario 4

Chapter 4

Experiment Methodology

4.1 Preparatory Phase

4.1.1 Pre-Survey

(For each sensor,dc and context show graphs from the pre-survey why each of them was chosen)

The pre-survey [1] is a survey created that runs before the deployment of the social experiment. This survey was made in order to study the perception of users on the three features to be studied:

1. Sensors - The mobile mobile sensor data shared
2. Stakeholders - The entity to whom mobile sensor data is shared
3. Context - The purpose for which mobile sensor data is shared

From each of the above mentioned features, there were a lot of sub-features to choose from. Sub-features are the elements that come under Feature. For example, Light sensor is a sub-feature of Sensors and Corporation is a sub-feature of Stakeholders. Increasing the number of sub-features for each feature in the experiment in turn increases the number of data requests posed to the user. Additionally, we wanted to gain insight into the perception of users on the three features. Hence the survey was prepared to understand the above(link to appendix). Also, it can help us redesign some of the aspect of the experiment based on the ambiguities found and user feedback. The participants pool consist of both people who are aware and unaware of data privacy and sensors. Participants were not paid for filling out the survey.

[1] https://descil.eu.qualtrics.com/SE/?SID=SV_0xGS6kfmr8GtQd7

4.1.2 Sub-Features

194 pre-survey entries have been filled out by the participants. Using the data obtained from the pre-survey, four sub-features were chosen from each feature.

4.1.3 Privacy Options

Each data request is accompanied with privacy options ranging from 1 to 5. Option 1 indicates that the users would like to share their raw data without any sort of summarization or filter. 5 indicates that the users would not like to share data for this data request. The options in between have linearly scaled summarization levels assigned to them ranging from least privacy (1) to most privacy (5). For more information refer to 3.2.8.

4.1.4 Question Structure

A data request is when a stakeholder asks users for mobile sensor data for a particular context or purpose. From now on, we will refer to mobile sensor data as just data. Each data request to the user is posed in the form of a question with the following template :

”Please choose the amount of X sensor type data shared with Y stakeholder for use in a Z context app”

where Sensors X can be :

1. Accelerometer
2. Noise
3. Location
4. Light

where Stakeholders Y can be:

1. Corporation
2. Educational Institution
3. Non Governmental Organization
4. Government

and where Contexts Z can be:

1. Environment
2. Health/Fitness
3. Navigation

4. Social Networking

In total this makes 64 data requests to the user.

4.1.5 Budget and Experiment Duration

The experiment is set to run for a total of two days, excluding the time taken for the entry phase and exit phase. The budget set for the core phase of the experiment is $B = 35$ Chf and is excluding the cost of participation in the entry and exit phase. Participants are paid 10 Chf for coming to the Entry Phase, and 15 Chf for participating in it. Similarly for the Exit Phase, participants are given 10 Chf for showing up, and 5 Chf for participating in it. Out of the budget B, $\frac{1}{7}$ is given away for the participation of the users in the core phase.

4.2 Entry Phase

explanation of screen shots

4.2.1 Collecting General User Information

As the figure 4.1 shows, the users are asked to answer some personal non-intrusive questions. The following is asked from the users:

1. Gender
2. Employment Status
3. Education Level
4. Year of birth
5. Country where user has lived most of his life
6. How many time a day do you check your Mobile phone per day.
7. Kind of applications the user has in the mobile phone.

The users may go back and re-answer the questions, but once submit button is pressed in the screen 4.1c, the data is sent to the server and hence cannot be changed. Users cannot navigate to the next pages without filling out all the questions.

4.2.2 Categorization of Features

As described in chapter 3, the users need to categorize the features Sensors, Stakeholders and Contexts. As shown in figure 4.2a, the features are indicated followed by 5 options of privacy ranging from ”very low privacy intrusion” to ”very privacy high privacy intrusion”. The option ”very low

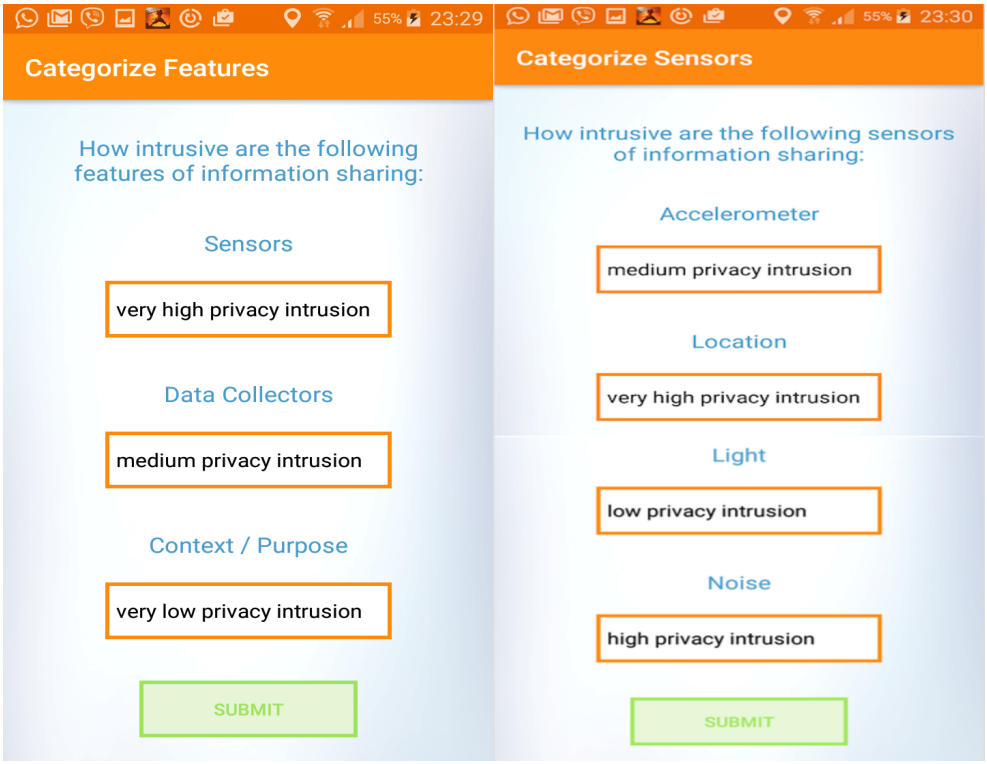
(a) User Information Screen 1 (b) User Information Screen 2 (c) User Information Screen 3

Figure 4.1: User Information Screens

privacy intrusion" means that the feature does not affect the users mobile sensor data sharing decision, whereas "very privacy high privacy intrusion" refers to a feature that very much affects the sharing of mobile sensor data. Users need to click on the drop down menu to choose one of the privacy intrusion options. All options are compulsory, and no default option is provided. Users cannot navigate to the next page without filling out all the questions.

4.2.3 Categorization of Sub-Features

For each of the features categorized in the previous sub-section, each of the sub-features need to be categorized in a similar fashion. Again, the privacy options range from very low privacy intrusion" to "very privacy high privacy intrusion" like in section 4.2.2 . The users are first presented with the



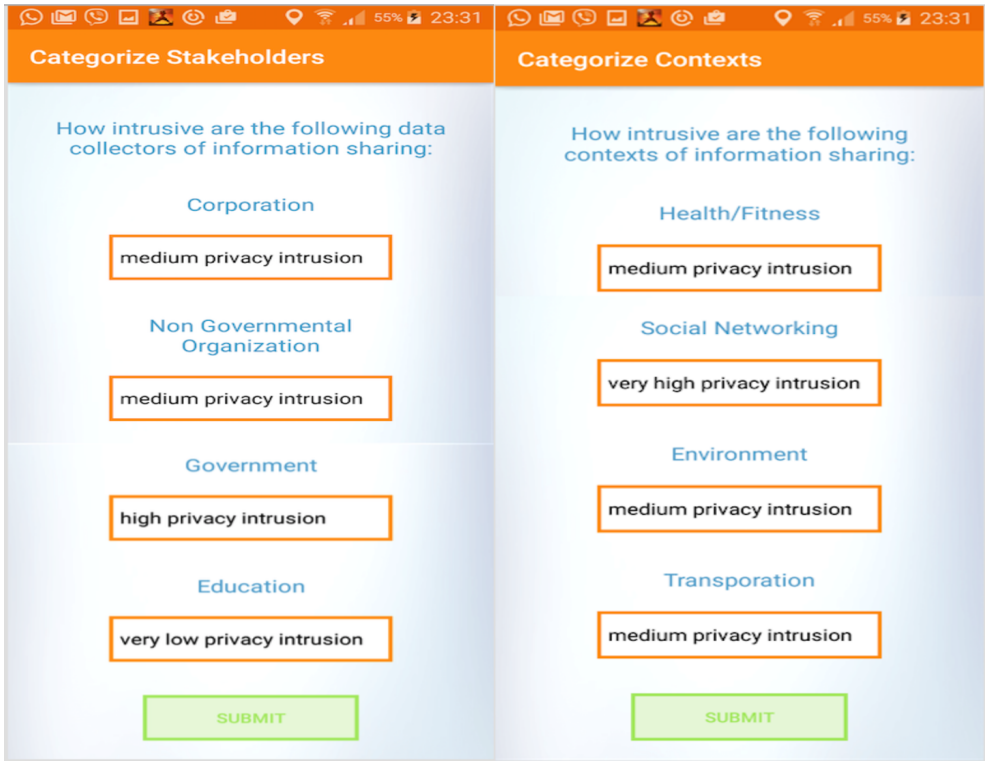
(a) Categorizing Features (b) Categorizing Sensors

Figure 4.2: Categorizations

categorization of Sensors sub-features as shown in figure 4.2b. Below each sensor is a drop down menu where the user can choose how much each of the sensors would affect the mobile sensor data sharing. Once all the sensors have been associated with an privacy intrusion level, the user can click the green submit button and is directed to the next page where the sub-features of stakeholders need to be in turn categorized in a similar fashion. This is depicted in figure 4.3a.

Each stakeholder type has a drop down menu each where the user can again classify how much each of them affect data sharing. Once the user has finished entering the privacy intrusion level for stakeholders, the user can click the green submit button and is directed to the next page. On this page, the user will need to categorize how much each of the Context's sub-features affect mobile sensor data sharing. This is depicted in figure 4.3b. Each context has a drop down menu below, where the user can rate each context from low privacy intrusion" to "very privacy high privacy intrusion". Once this has been done the user can click on the green submit button.

The user will be redirected to the next page only if all the drop down boxes



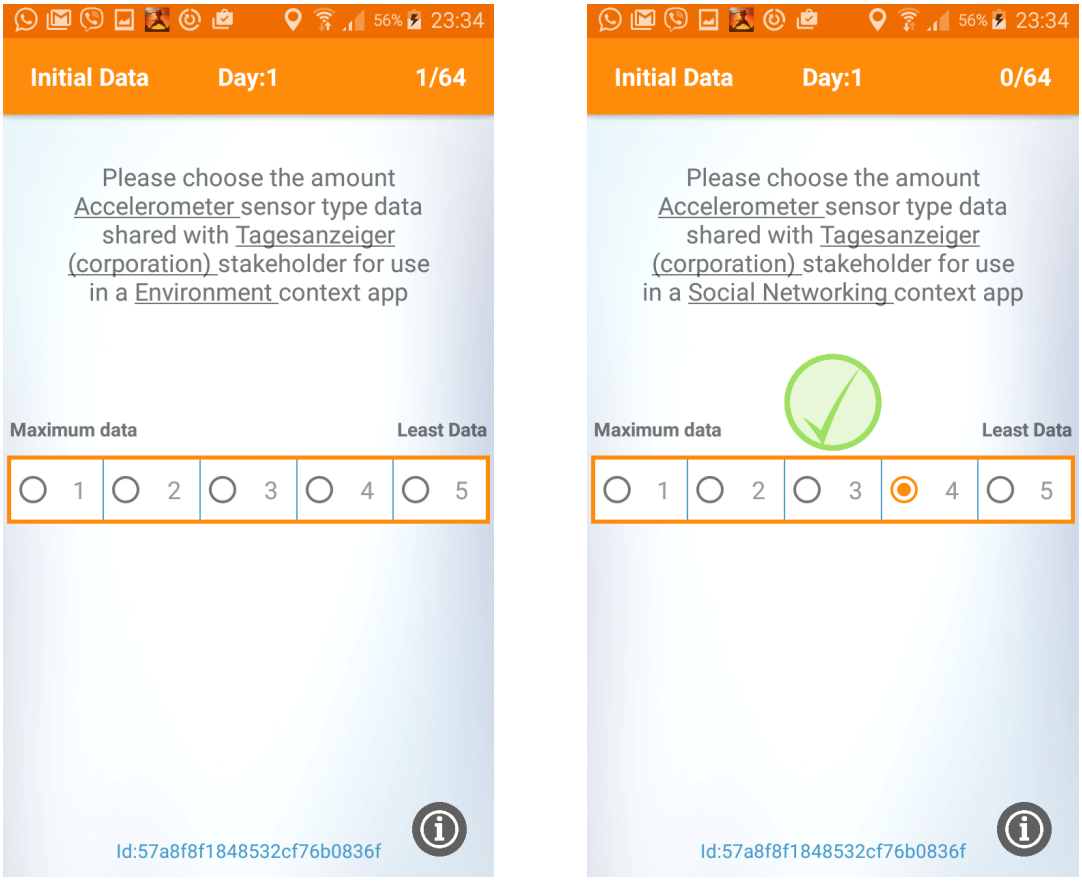
(a) Categorizing Stakeholders (b) Categorizing Contexts

Figure 4.3: Categorizations

have been filled out. All questions are compulsory there is no default choice.

4.2.4 Answering Questions with No Incentives

After the categorization questions are answered and user answers recorded, users will be presented with 64 questions. Each question is a mobile sensor data request to the users. Users can choose from the available five privacy options mentioned in section 4.1.3. The options are indicated as a measure of how much data users can give, ranging from maximum data to least data. The higher the privacy of the option, the less is the sensor data given away for that request and vice versa. Users can change the answers to a data request until the green submit button on top of the options that appears is clicked. The screen with the data request is shown in figure 4.4a. After the users choose an option for the data request, a green submit button appears which is shown in figure 4.4b. After clicking on the submit button, response to the data request cannot be changed. At this time, no indications of credit gain or privacy improvements are indicated.



(a) Question Screen

(b) Question Screen with Submit button

Figure 4.4: First Day Screen

The *"i"* button at the bottom right of the screen is clickable. This takes the user to the FairDataShare portal. Figure 4.5b shows the homepage of the portal. Users need to then click on the data generator registration section of the website where users can signup with their:

1. Username
2. Password
3. Email
4. Unique Identifier

The unique identifier is located at the bottom of the page is an alphanumeric sequence. If it is long pressed the user can select the identifier, then copy and paste it in the textbox asking for the unique identifier. Figure 4.5b shows what the registration page looks like. The users can use this website to see

all the data collected from them for all the mobile sensors. More details about the FairDataShare portal refer to the section 4.5.

(a) FairDataShare Homepage

(b) Data Generators Registration Page

Figure 4.5: FairDataShare Portal

In the task-bar, the user can see the bidding day number and how many questions have been answered from the total available. Day number one corresponds to the day where users answer questions with no incentives of any kind. Once all the questions have been answered, the user goes to the core phase of the experiment, which starts at day number two.

4.3 Core Phase

Once the entry phase is done, the user is presented with the screen shown in figure **??**. The first presented screen after the entry phase is over is what is called the ”improvement screen”. The button numbered 3 represents ”improve privacy” and the button numbered 4 represents ”improve credit” respectively. The items numbered 6 and 5 represent the privacy and credit obtained by the user respectively. Privacy is measured in terms of the percentage of mobile sensor data not traded to the stakeholders. Credit is measured in terms of the currency Swiss Francs. The button numbered 9 is the button that takes the user to the FairDataShare portal. The user can login into the portal after a minimum of 24 hours after the start of the core phase to see the data that has been collected and shared with the stakeholders. The item numbered 8 is the unique identifier of the user. This can be selected and copied by long pressing the unique identifier for one second. The item numbered 7 is the round number which indicates the number of times the user has answered all the data requests. The item numbered 2 is the number of questions the user has answered in the current round. Item number 1 indicates the experiment day number.

There are a total of 64 data requests, hence when all the 64 have been answered, the number of questions answered is reset and the number of round answered increases by one. This indicates all the data requests that have

been answered and how many are left unanswered.

Each question will have 5 options to choose from ranging from maximum data sharing to least data sharing.

From the starting time of the core phase till 24 hours later marks one bidding day. Once 24 hours is over, another bidding day starts where the privacy and credit metrics are reset. The day number in the task bar is incremented by one. The user has to answer all the data requests again for this new bidding day. Previous responses to data requests are not carried over to the next day. If a data requests is not answered, it is considered that the user does not want to trade mobile sensor data for that request. Additionally, each data request carries a participation fee, this is irrespective of the amount of mobile sensor data shared, by not participating in a data request the user foregoes this credit gain. The core phase goes on for a period of 48 hours.

4.3.1 Improve Privacy or Credit

The improvement screen shown in figure 4.6 is where users can choose whether he would like to improve the privacy or the credit that has been obtained. The elements of this screen have been explained in the previous section 4.3. The improve credit button should be chosen if the user is interested in maximizing the credit already obtained further. This uses algorithm that uses the previous user answers to put forth a data request that can increase the credit to the maximum. The credit improvement button is represented by the number 5. Similarly, the improve privacy button is used to further improve the privacy that has been obtained. This puts forth a data request that can further increase the user privacy. Then again, the ultimate change in the privacy or credit metrics depends on the option chosen by the user for the data request. The privacy improvement button is represented by the number 6.

Scenario example for each button is given in the next section after introducing the next screen.

For example, if a user chooses to improve the privacy, then clicks on improve privacy button and gets a data request, but still chooses option maximum data (least privacy) for that data request, this may not improve his privacy but decrease it. This is because option 1 indicates that the user trades all the data for this request. Trading all data gives the user more credit, but decreases the privacy metric.

Similarly, if a user chooses to improve the credit, then clicks on the improve credit button and gets a data request. Then the user chooses the option least data (maximum privacy) which indicates that no data is traded for this request, this is counters the initial desire to improve the credit obtained.

Trading no data increases one's privacy, but does not increase the credit to the maximum.

Therefore, an actual improvement in the chosen metric depends on the chosen improvement button chosen and the choice of the appropriate option for that data request.

4.3.2 Answering Questions with Incentives

After choosing a metric to improve, a screen is presented as shown in figure 4.7a. This screen is called the "bidding screen". This screen is very similar to the screen 4.6 presented in the entry phase, except that the user is aware of the amount of privacy and credit obtained. Additionally, the user can see information about how the privacy and credit will increase or decrease for each data request, according to the chosen option. The items numbered 11 are the possible answers ranging from one to five (option numbers are not indicated on the screen). The items numbered 12 are the improvement in privacy for each possible option of the current data request. The items numbered 13 are the improvements in credit for each possible options of the current question. Once the user decides on which options to choose according to how much data wants to be traded, the users can click on the radio option as explained in section **??** and then click again on the green submit button shown in 4.7b to confirm the answer. Once the green button has been clicked on, answers cannot be changed. The user has the possibility to go back to the improve screen from the bidding screen using the back button. Using the back button in the improve screen leads the user out of the application.

Additionally, for every question there is an orange recommendation box surrounding some options. This recommendation is highlighted in figure 4.8a. This gives an indication to the user as to which options can improve the privacy or the credit compared to the previous time the user has answered this data request. For example, if the user has previously answered option 3 to a data request and has clicked on improve credit, the system puts an orange box around options 1,2 and 3. Similarly, if the user clicked on improve privacy button, the system would recommend the options 3,4 and 5. Two examples of this are provided in figures 4.8.

4.4 Exit Phase

After the end of the core phase, the participants are asked to fill up a survey based on their experience of the experiment. Some questions are about the rewards received, the privacy and credit metrics, design of the application,

and how the experiment was conducted. The survey [2] is linked to the user using the unique identifier assigned in the application. Once the survey is filled, the users receive their money for the entry phase, core phase and exit phase together, but only if they did not have their phones switched off throughout the experiment and participated in the core phase. This is done by checking the data collected on the server.

4.5 FairDataShare Web Portal

The FairDataShare portal [3] is a website where users can view the data collected from them during the core phase of the experiment. Below is an explanation of how users and stakeholders can view mobile sensor data.

4.5.1 Data Generator's Portal

Users first register as data generators as indicated in the section 4.2.4.

Once the users are registered, they can come back to the portal after 24 hours period or later to view their mobile sensor data collected in the server. The data portal login page is shown in figure 4.9a. Since the users are already registered from the mobile phone in the entry phase, they can go to the portal from their computers and this time login instead of register. Users should enter their:

1. Username
2. Password

Once done, users will be redirected to the data collection page shown in figure 4.9b with the following options in the task-bar :

1. Accelerometer
2. Light
3. Noise
4. Location

Users can choose the sensor from the task-bar whose data they want to see by clicking on it. The data displayed includes the following columns :

1. Timestamp
2. Bidding day
3. Sensor Values

[2] https://descil.eu.qualtrics.com/SE/?SID=SV_3POySMqNeOO6v5j
[3] http://fair-data-share.inn.ac/

Figures 4.10a, 4.10b, 4.11a and 4.11b show examples of data that can be seen for the location, light, accelerometer and noise sensor.

In the experiment, day number one is the entry phase, the core phase is day number two and three.

4.5.2 Stakeholder's Portal

For a stakeholder to view data, they need to register in the portal shown in figure 4.5a by clicking register. Once that is done, the page in figure **??** is shown asking for :

1. Company Name
2. Email
3. Stakeholder Category
4. Company Website

Stakeholder category is the type the stakeholder comes under such as :

1. Corporation
2. Educational Institution
3. Government
4. Non-Governmental Organization

After this, the stakeholder can click on the register button. Once registered, the stakeholder can login like in 4.12b. Access is then granted to the page in figure 4.13. The stakeholder can choose from each drop down list:

1. A sensor
2. A context
3. An anonymous user
4. A bidding day number

Once this is entered, the stakeholder can see data for that user with the privacy level decided by the anonymous user. If the stakeholder does not see any data, it means the user did not share data for this request. Stakeholders can view sensor data in a similar fashion to users shown in figures 4.10 and 4.11.

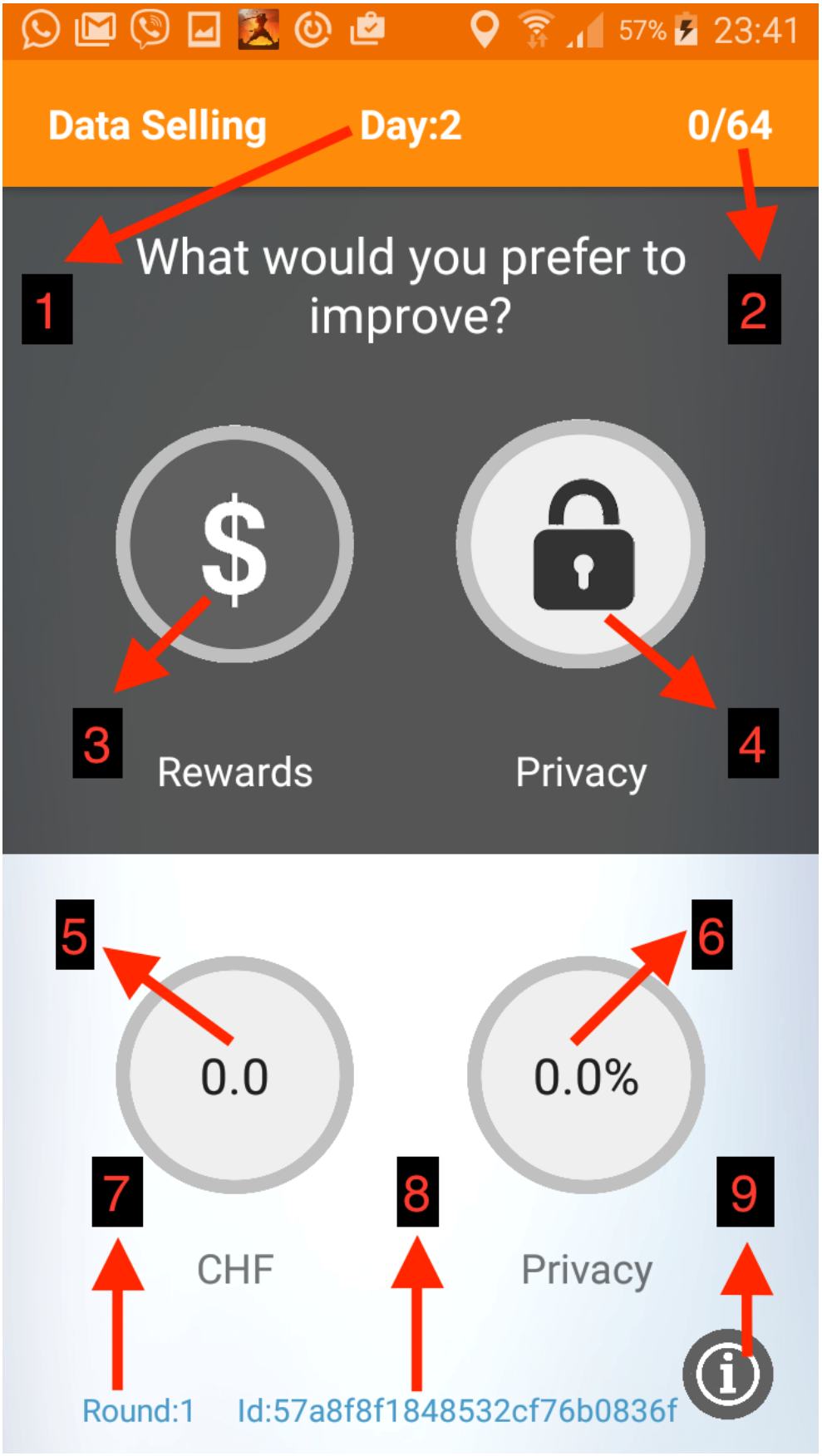
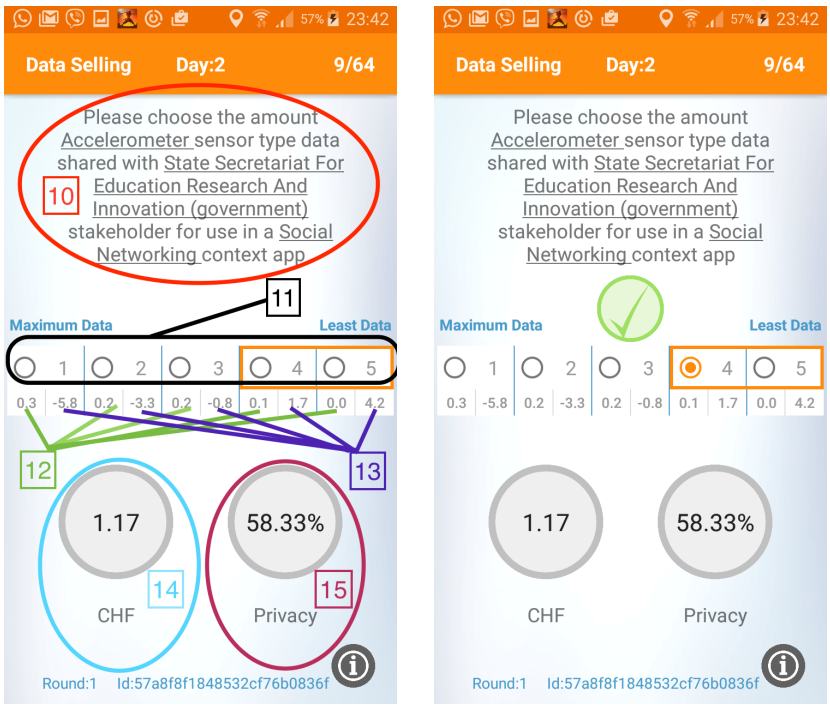


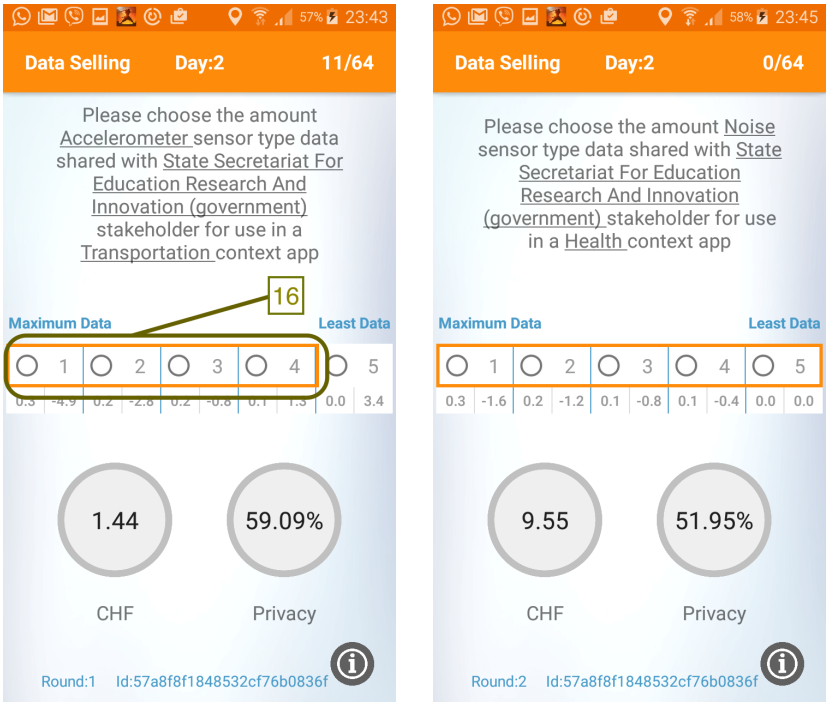
Figure 4.6: Improvement screen



(a) Bidding Screen

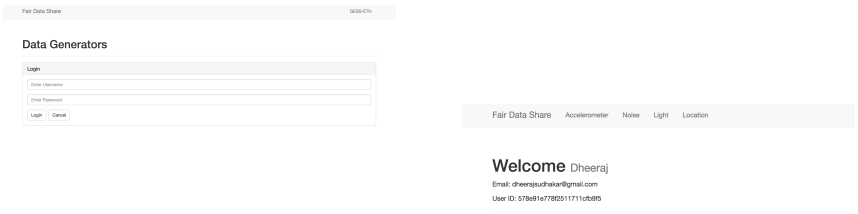
(b) Bidding Screen with Submit Button

Figure 4.7: FairDataShare Portal



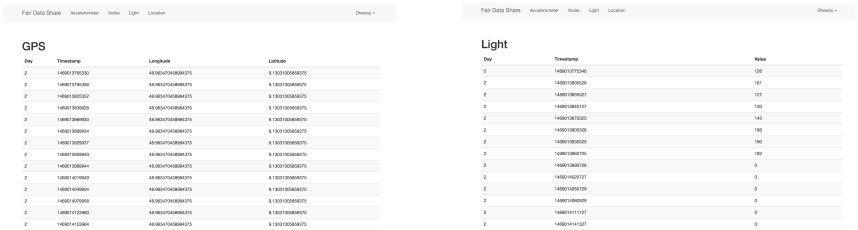
(a) Example 1 (b) Example 2

Figure 4.8: Recommendation Box



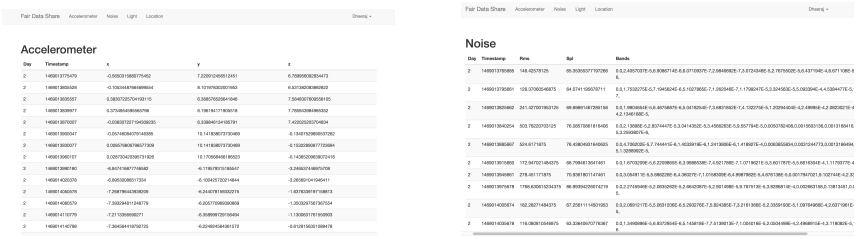
(a) Login Page (b) Welcome Page

Figure 4.9: Entering the Portal



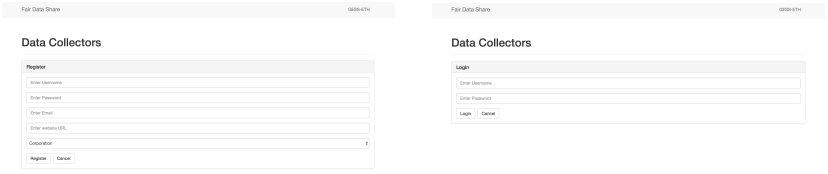
(a) Location Data (b) Light Data

Figure 4.10: User Data



(a) Accelerometer Data (b) Noise Data

Figure 4.11: User Data



(a) Registration Page (b) Login Page

Figure 4.12: Entering the Portal for Data Collectors

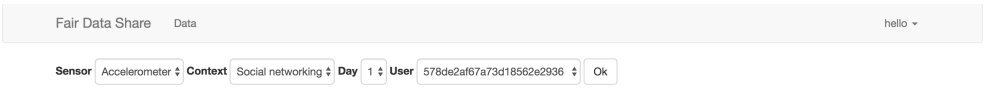


Figure 4.13: Data Collectors Welcome Page

Chapter 5

Explanation of the Mobile Application

5.1 The Building Blocks

The following sections will explain integral parts of the server and client part of the mobile application. A gist of the architecture is shown in figure 5.1. As it can be seen, the mobile app represents the user participating in the experiment. As the experiment goes on, mobile sensor data and responses to the data requests which are collected are periodically sent to the Kinvey Data Store. The users can choose to login into the FairDataShare Portal from their computer or the mobile app. Once the user is authenticated, the user request is sent from the FairDataShare server to the Kinvey Data Store. Kinvey in turn fetches the appropriate data and gives it to the FairDataShare Server. This in turn structures the data so it can be easily readable, and pushes it to the user to see on the portal. The concept is similar for the Stakeholders, except they can only access the portal through the computer and not the mobile app.

5.2 The Mobile Application

5.2.1 Local Storage

The local storage is an integral part of the application. The database used is SQLite and is the default database for the Android environment. Small sized unrelated data is stored in preferences files, whereas larger related data is stored in the database. The following paragraphs will explain each database present in this application followed with their function. All tables explained here are pertaining to the user using the mobile application.

Figure 5.2a shows the QUESTIONSTORE's table schema. This table stores each possible data request with its sensor $SENSOR$, stakeholder $STAKEHOLDER$ and context $CONTEXT$. Each of these are represented by an integer, for ex-

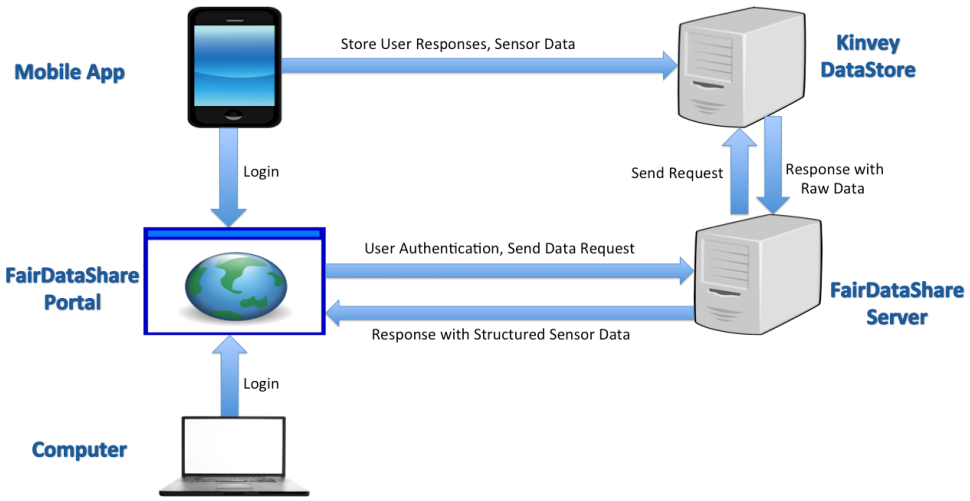
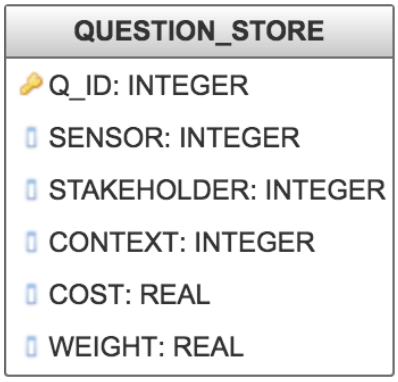
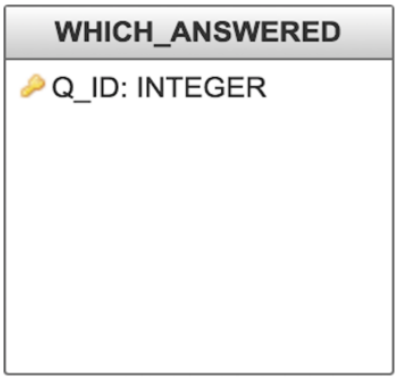


Figure 5.1: Conceptual Diagram of Mobile Application Architecture



(a) Table Schema of QUESTION-STORE



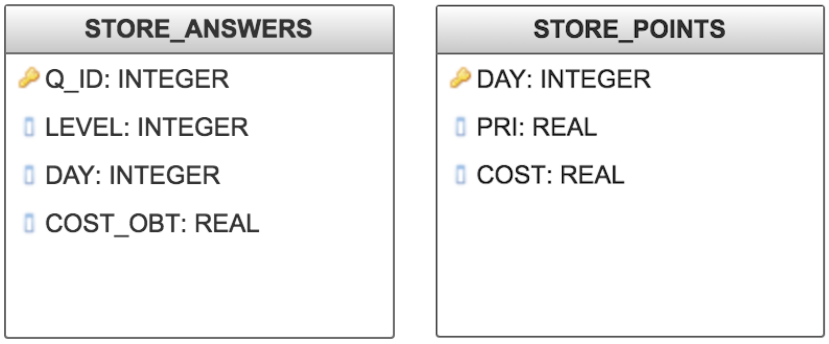
(b) Table Schema of WHICHANSWERS

Figure 5.2: Table Schemas

ample sensor 0 stands for Accelerometer sensor. Each data request is accompanied by an unique question identifier QID, weight assigned $WEIGHT$ and the cost $COST$. This data is not sent to the server.

Figure 5.2b depicts the table WHICHANSWERS's table schema. This stores the questions identifier QID of each data request that has been answered by the user for each round. This is helpful while fetching data requests, so as not to fetch the request twice in the same round. This makes sure that all questions are answered before answering them for a second time. This data is not sent to the server.

Figure 5.3a explains the schema of STOREANSWERS table. This table is



(a) Table Schema of STOREANSWERS

(b) Table Schema of STOREPOINTS

Figure 5.3: Table Schemas

used to store the data request identifier QID with the corresponding user responses $LEVEL$, along with the increase or decrease in credit obtained $COST_OBT$. The total cost can be calculated by adding all the costs in this table. Similarly, the total privacy can be calculated by averaging all the user responses in this table. Only the most recent responses are stored in this table. This content is not sent over to the server.

Figure 5.3b denotes the schema of STOREPOINTS table. This table is used to store the credit and privacy obtained for each bidding day. This information is sent to the server as soon one bidding day is over.

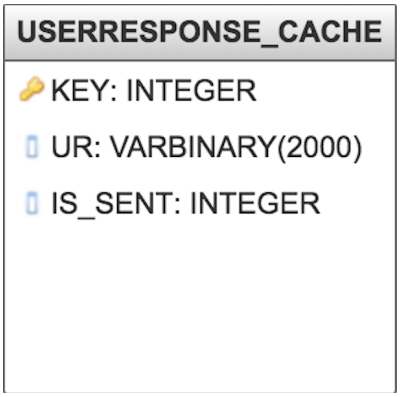


Figure 5.4: Table USERRESPONSECACHE Schema

Figure 5.4 depicts the USERRESPONSECACHE tables's schema. This table stores a unique key KEY for each user response, followed by a flag $ISSENT$, which is 1 if the response is not sent to the server, and 0 if it is sent. The user response saved consists of the following entries :

1. User Id
2. Timestamp of response
3. Sensor Id
4. Stakeholder Id
5. Context Id
6. Privacy Level answered for this data request
7. Cost obtained for this data request
8. Current Total Privacy of user
9. Current Total Credit of user
10. Maximum Obtainable Credit for this data request in this round
11. Metric Chosen to Improve (Improve Privacy or Improve Credit)

All of the above fields are packed into the field *ur* shown in 5.4. The data in this table is sent to the server. Once the entry is sent to the server, the *ISSENT* field is changed to 0 and deleted locally. The unique keys *KEY* are useful for deleting sent entries.

Figure 5.5 and 5.6 show the table schemas for data storage of the following sensors:

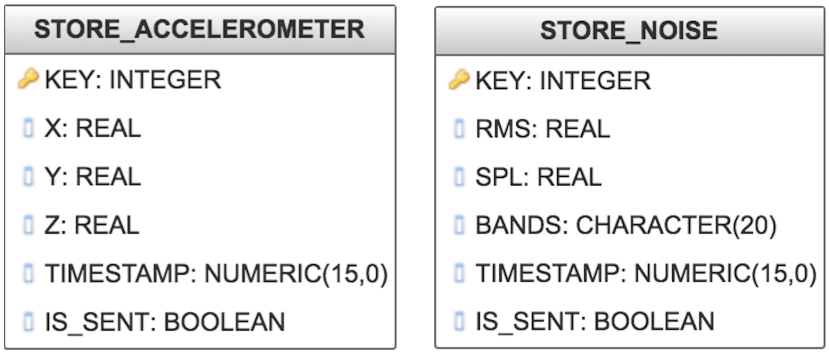
1. Accelerometer in the STOREACCELEROMETER table
2. Noise in the STORENOISE
3. Location in the STORELOCATION
4. Light in the STORELIGHT

The general schema for all the sensors is the following :

1. *KEY* - Uniquely identifies each sensor entry
2. *TIMESTAMP* - The time the sensor value was collected
3. *ISSENT* - Denotes whether the sensor entry has been sent to the server or not
4. The other columns are specific to each sensor and represent the actual sensor values of the user collected

5.2.2 Alarms and Notifications

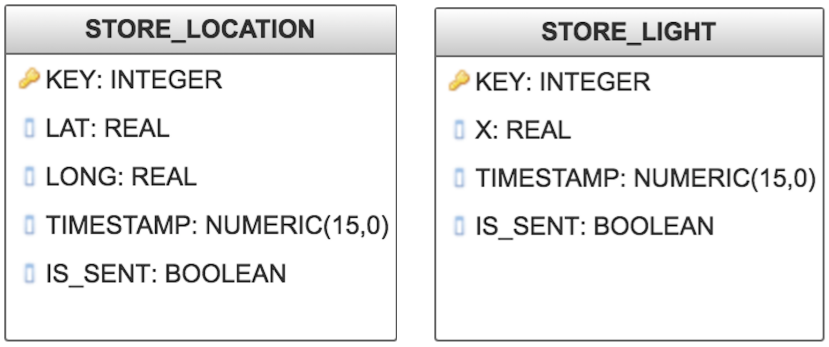
Every bidding day where the user can answer data requests lasts for a period of 24 hours. After one bidding day is over, the system needs to be informed



(a) Table Schema of STOREACCELEROMETER

(b) Table Schema of STORENOISE

Figure 5.5: Table Schemas for Sensor Data



(a) Table Schema of STORELOCATION

(b) Table Schema of STORELIGHT

Figure 5.6: Table Schemas for Sensor Data

in a timely to perform some application critical functions. The function performed are explained in detail in section 5.2.2. To inform the system of such an event Android provides the functionality in the form of alarms.

Alarms can be set to go off just once or in a repeated fashion to trigger tasks. Unfortunately, the alarms provided by Android are not exact for some versions, in the sense that they are triggered around that time set but not exactly to optimize the battery. Hence, we decided to set the repeating alarms manually.

The first time the alarm is set to ring in exactly 24 hours, but things change when the phone is switched off. One of the conditions of the experiment is not to have the phone switched off at any time. Neverthless, we take into account the scenario where the phone is kept switched off for a period of time. There are various things that can happen:

1. The phone is rebooted.
2. The phone is switched off, during this time an alarm is missed.
3. The phone is switched off for a period greater than 24 hours. One or more alarms can be missed.

Once the phone is switched off, all alarms are erased. Alarms do not execute when the phone is switched off. Hence, when the phone switches on, BootReceiver service of the application is triggered with pseudocode 1.

Algorithm 1 BootService Algorithm

```
1: procedure BootService
2:     now ← current timestamp
3:     i ← timestamp of last triggered alarm
4:     if now − i < 86400 then
5:         Call SetAlarmLater()
6:     else
7:         Set alarm in 200 seconds
```

This checks whether an alarm has been missed, if it has 200 seconds is given for the phone to stabilize before triggering it. Otherwise, a new alarm is set using pseudocode 2. To set an alarm we need the time difference between now and when the alarm should ring. After that is calculated, the alarm is set.

Algorithm 2 Alarm Algorithm

```
1: procedure SetAlarmLater
2:     now ← current timestamp
3:     i ← timestamp of last triggered alarm
4:     latertime ← i + 86400
5:     latergap ← latertime − now
6:     Set Alarm in latergap seconds
```

Going to the Next Data Sharing Day

Once the alarm rings, it marks the end of a bidding day. Once a bidding day ends a number of tasks need to be executed and for this the NextDayService is triggered, which is described in pseudocode 5. To start with the the privacy and credit is sent to the sent to the server and stored locally in the STOREPOINTS table. *Privacy* which is the total privacy obtained, *Credit* is the total credit obtained, *Round* which is the number of time the user answered all the questions and *CurrentQuestion* which is the current question

the user is answering is all reset to zero. The *Day* corresponds to the current day number is incremented by one to denote the next bidding day.

Algorithm 3 NextDayService Algorithm

```
1:  procedure NextDayService
2:      Store Privacy, Credit, Day in STOREPOINTS
3:      Send Privacy, Credit, Day to Server
4:      Privacy, Credit, Round, CurrentQuestion ← 0
5:      Day ← Day + 1
6:      Store current time
7:      Call Summarization()
8:      if Day > End then
9:          End experiment
10:     else
11:         Update user interface elements
```

The current time of executing the alarm is saved in case the phone is rebooted or switched off. After that, the sensor data which is saved locally needs to be summarized, the corresponding method is called and is explained in pseudocode 4. Finally we need to check if the experiment is over or not and update the user interface accordingly. This means either we update the various metrics on the improvement and bidding screens, or we show the end of experiment screen.

5.2.3 Fetching Data Requests

A data request needs to be fetched in two scenarios :

1. After a question has been answered in the first bidding day.
2. After the privacy or credit improvement button has been clicked.

In the first bidding day, once a data request has been answered the next one is fetched sequentially from the database. This just requires knowing the current data request number and fetching the next data request from table QUESTIONSTORE.

For the other bidding days, fetching of the data requests depends on the improvement button chosen. According to the choice, the following is done:

1. Improve Privacy - Obtain question from table STOREANSWERS where user has answered with lowest privacy
2. Improve Credit - Obtain question from table STOREANSWERS where user has answered with highest privacy

In addition to sending the data request to the user interface, we need to show how choosing each option of the data request will affect the total privacy and total credit metrics. To do this for the total cost, we output $last - possible$, where *last* stands for the credit obtained the last time the data request was answered. *possible* stands for the maximum amount of credit that can be obtained for this option (each data request has five privacy options). The possible total cost changes are shown under the options. For more detail on how credits are split among options in a data request refer 4.1b.

Each option of a data request has an associated percentage of data that is given away as described in 4.1b. According to the percentage of data given away, the total privacy is calculated for each possible option. The difference between the current privacy and each possible total privacy is calculated and indicated under each option. This gives an indication to the user as to what each option will do to the metrics.

5.2.4 Recording User Choices

5.4 describes the table USERRESPONSECACHE. Each time a user enters a response to a data request, all the fields mentioned in section 5.2.1 are recorded and store in a class object. This object is transformed into a byte array so as to be stored easily in the table as is. When the JobNetworkService described in 5.2.6 is called, the class object is sent as is to the server.

5.2.5 Sensor Data Collection and Summarization

Sensor data is collected from the following sensors :

1. Accelerometer sensor
2. Noise sensor
3. Location sensor
4. Light sensor

A sensor service is triggered when the application is installed and is stopped when the experiment is over. This collects data from every sensor every 30 seconds and stores it in the appropriate tables mentioned in section 5.2.1. At the end of a bidding day, sensor data needs to be summarized according to the wishes of the user. This starts by first finding out the lowest privacy level for each sensor. Privacy levels range from one to five, that is from the lowest to highest privacy levels. Using this level summarization is done as shown in pseudocode 4. Each privacy level corresponds to an action:

1. 1- All data is sent to the server
2. 2- Send 75% of the data

3. 3- Send 50% of the data

4. 4- Send 25% of the data

5. 5- Do not send any data

Initially all the sensor data has a field $ISSENT$ with value of zero. Data that should be sent to the server is set with $ISSENT = 1$, and all that have value $ISSENT = 0$ are ignored.

Algorithm 4 Summarization Algorithm

```
1:  procedure SUMMARIZATION
2:      for each sensor do
3:          Fetch sensor data from sensor table
4:          level ← Fetch user privacy level
5:          if level ← 1 then
6:              Set all ISSENT ← 1
7:          else if level ← 2 then
8:              for 3 out of every 4 records do
9:                  ISSENT ← 1
10:         else if level ← 3 then
11:             for 1 out of every 2 records do
12:                 ISSENT ← 1
13:         else if level ← 4 then
14:             for 1 out of every 4 records do
15:                 ISSENT ← 1
16:         Delete all entries with ISSENT ← 0
17:         Update Database
```

5.2.6 Server Synchronization

User responses and sensor data need to be sent to the server. This is done periodically every 5000 seconds in order to free up space on the phone when the internet is available. This is triggered first when the application is started. Data is fetched from the tables in the database. Data with fields marked as $ISSENT = 1$ is data that is ready and that has not been sent yet to the server. Such data is sent, and when an acknowledgement is received, this data is deleted from the table.

Algorithm 5 JobNetworkService Algorithm

```
1:  procedure NetworkService
2:      Fecth data from USERRESPONSECACHE
3:      for each record do
4:          if ISSENT == 1 then
5:              Send record to Server
6:              if SUCCESS then
7:                  Delete record
8:      for each sensor do
9:          Fecth data from sensor table
10:         for each record do
11:             if ISSENT == 1 then
12:                 Send record to Server
13:                 if SUCCESS then
14:                     Delete record
```

5.3 The Server

5.3.1 Kinvey Data Storage

Kinvey [1] is a mobile backend as a service which provides a platform for mobile phones to link applications to a backend cloud storage. For the purpose of this application the backend has been used to store data for some business logic implementations.

Security

All communication from the application to the server is encrypted using TLS/SSL encryption [2]to communicate with the backend service. This is automatically provided by the Kinvey SDK.

Collection Store

Locally, all information is store in SQLite which is a relational database. The database used in Kinvey is MongoDB so instead we have collections on the server. When the user starts the application, general personal information is entered as explained in 4.1b. This data is stored in the collection UserInformation with the schema shown in the screen shots 5.7 and 5.8.

Once this is done, users have to categorize the various Features, Sensors, Stakeholders and then the various Contexts. This information is sent to the

[1] http://kinvey.com/

[2] Kinvey white paper : KINVEY CLOUD SERVICE: SECURITY OVERVIEW 2014

birth_year	check_mobile_frequency	country	education	education_backgrou…	education_level	employment_status	entertainment	finance
1994	3	"France"	0	0	3	6	0	1
1924	3	"Arménie"	0	0	4	4	0	0
1923	3	"Armenia"	0	0	4	2	0	0
1922	3	"Aruba"	0	0	3	2	0	0
1992	1	"France"	0	0	4	6	0	0
1991	1	"France"	0	0	5	6	0	0
1924	3	"Argentin…	0	0	3	2	0	0
1921	3	"Andorre"	0	0	2	1	0	0
1923	3	"Argentin…	0	0	2	2	0	0
1922	3	"Antigua-…	0	0	2	1	0	0
1922	3	"Anguilla"	0	0	2	2	0	0
1922	3	"Anguilla"	0	0	2	1	0	0
1923	3	"Angola"	0	0	2	2	0	0
1926	2	"Angola"	0	0	4	6	0	0
1924	3	"Andorre"	0	0	3	5	0	0
1924	3	"Angola"	0	0	4	6	0	0
1920	5	"Fiji"	0	0	1	3	0	0

Figure 5.7: Screenshot of Collection UserInformation Part 1

gender	health	medical	mobile_sensor_priv…	music	user_id	navigation	news	productivity	shopping	social_network
2	1	0	3	1	"57a8f8f1848532cf7…	0	0	0	1	1
2	0	0	3	0	"579a148f352257bc0…	0	0	0	0	0
2	0	0	3	1	"57975541e813f9973…	0	0	0	0	0
2	0	0	2	0	"57975159890927b61…	0	0	0	0	0
2	0	0	3	1	"57935b55a67b0ba32…	1	0	0	0	0
2	1	0	3	1	"579357faa67b0ba32…	1	0	0	0	1
2	1	0	3	0	"579357faa67b0ba32…	0	0	0	0	0
1	0	0	3	0	"57931cad866a46bd5…	1	0	0	0	0
2	1	0	3	0	"57930d55837af5db6…	0	0	0	0	0
1	1	0	3	0	"5792471c493006891…	0	0	0	0	0
2	1	0	3	0	"57923fbfc3d7cee30…	0	0	0	0	0
2	1	0	3	0	"57923946c3d7cee30…	0	0	0	0	0
2	0	0	3	0	"5792373d3692318e3…	1	0	0	0	0
2	0	0	4	1	"57922e1afb5591741…	0	0	0	0	0
2	1	0	3	0	"57922a329bb316492…	0	1	0	0	0
2	0	0	3	1	"579224ffba4636590…	0	0	0	0	0
2	0	0	1	0	"5791d082bb71b5202…	0	0	0	0	0
1	1	0	4	1	"578e91e778f251171…	1	1	0	0	1

Figure 5.8: Screenshot of Collection UserInformation Part 2

server in collections named Features, Sensors, Stakeholders and Contexts. This is shown in 5.9, 5.10, 5.11 and 5.12 respectively.

All the data stored locally on the mobile phone which is sent by the JobNetworkService explained in section 5.2.6 is received by Kinvey. User responses are store in collection USERRESPONSE shown in 5.13 and 5.14.

The sensor data sent by the JobNetworkService is stored in collections named after the sensors themselves. The schema of the tables is shown in figures 5.15, 5.16, 5.18 and 5.17.

user_id	context	data_collector	sensor
"57a8f8f1848532cf7…	1	3	5
"579a148f352257bc0…	2	3	3
"57975541e813f9973…	4	5	1
"57975159890927b61…	4	3	1
"57935b55a67b0ba32…	1	3	5
"579357faa67b0ba32…	1	3	5
"57931cad866a46bd5…	3	2	2
"57930d55837af5db6…	4	1	2
"5792471c493006891…	4	2	2
"57923fbfc3d7cee30…	4	1	2
"57923946c3d7cee30…	3	1	2
"5792373d3692318e3…	4	1	2
"57922e1afb5591741…	4	2	2
"57922a329bb316492…	3	2	4
"579224ffba4636590…	4	2	2
"578e91e778f251171…	3	5	4
"578e28687d1cdd1b6…	4	3	2

Figure 5.9: Screenshot of Collection Features

user_id	acc	gps	light	noise
"57a8f8f1848532cf76b0836f"	3	5	2	4
"57a8f8f1848532cf76b0836f"	3	5	2	4
"579a148f352257bc0612c70b"	2	2	4	3
"57975541e813f99735dd0598"	2	4	3	3
"57975159890927b613d0f49f"	1	3	5	3
"57935b55a67b0ba32f81eeac"	3	5	1	5
"579357faa67b0ba32f81e724"	3	5	1	5
"57931cad866a46bd554a1896"	2	2	3	4
"57930d55837af5db6734a438"	2	1	2	4
"5792471c493006891e4e0c2b"	2	2	3	4
"57923fbfc3d7cee306b50957"	2	3	3	4
"57923946c3d7cee306b4fb44"	2	4	2	4
"5792373d3692318e3ba9e929"	2	1	2	4
"57922e1afb55917415770d44"	2	3	4	4
"57922a329bb316492f70242b"	2	1	2	4
"579224ffba46365901e66e78"	2	2	2	4
"578e91e778f2511711cfb9f5"	1	5	1	3

Figure 5.10: Screenshot of Collection Sensors

To keep track of all the existing users in the experiment, the collection Users stores all unique user identification strings. This is shown in 5.19.

Finally, the collection Score shown in 5.20 stores the total privacy, total credit obtained by the user for each bidding day.

user_id	corp	edu	gov	ngo
"57a8f8f1848532cf76b0836f"	3	1	4	3
"579a148f352257bc0612c70b"	3	3	4	2
"57975541e813f99735dd0598"	1	4	1	3
"57975159890927b613d0f49f"	2	4	4	5
"57935b55a67b0ba32f81eeac"	5	3	5	5
"579357faa67b0ba32f81e724"	5	3	5	5
"57931cad866a46bd554a1896"	2	4	4	1
"57930d55837af5db6734a438"	3	3	1	2
"5792471c493006891e4e0c2b"	2	4	1	1
"57923fbfc3d7cee306b50957"	3	4	3	2
"57923946c3d7cee306b4fb44"	3	3	3	3
"5792373d3692318e3ba9e929"	3	3	2	2
"5792373d3692318e3ba9e929"	3	3	2	2
"57922e1afb55917415770d44"	4	3	2	1
"57922a329bb316492f70242b"	2	4	2	1
"579224ffba46365901e66e78"	2	4	2	1
"578e91e778f2511711cfb9f5"	5	2	5	4

Figure 5.11: Screenshot of Collection Stakeholders

user_id	environment	health	social_networking	transportation
"57a8f8f1848532cf76b0836f"	3	3	5	3
"579a148f352257bc0612c70b"	4	2	2	2
"57975541e813f99735dd0598"	2	2	4	2
"57975159890927b613d0f49f"	3	1	5	5
"57975159890927b613d0f49f"	3	1	5	5
"57975159890927b613d0f49f"	3	1	5	5
"57935b55a67b0ba32f81eeac"	1	1	1	1
"579357faa67b0ba32f81e724"	3	1	5	3
"57931cad866a46bd554a1896"	3	3	3	5
"57930d55837af5db6734a438"	3	2	1	3
"5792471c493006891e4e0c2b"	1	2	1	3
"57923fbfc3d7cee306b50957"	3	3	2	5
"57923946c3d7cee306b4fb44"	3	2	3	3
"5792373d3692318e3ba9e929"	2	3	1	3
"5792373d3692318e3ba9e929"	2	3	1	3
"5792373d3692318e3ba9e929"	2	3	1	3
"5792373d3692318e3ba9e929"	3	3	2	4

Figure 5.12: Screenshot of Collection Contexts

Bussiness Logic

?? Most of the bussiness logic used for the FairDataShare portal is present on Kinvey. There are two main scripts stored in Kinvey:

1. Script to find the privacy preference

contexts	credit	credit_can_be	credit_gain	credit_question	data_collectors	timestamp	day_no
0	6.510009765625	0	-0.076904296874999…	0.17944335937499994	0	"2016-07-25 11:51:57.234"	3
1	6.436767578125	0.30029296874999994	0	0.35034179687499994	2	"2016-07-25 10:19:00.938"	3
0	6.436767578125	0.30029296874999994	0	0.35034179687499994	2	"2016-07-24 14:53:15.508"	3
2	6.436767578125	0.30029296874999994	0	0.35034179687499994	1	"2016-07-24 14:53:04.694"	3
3	6.436767578125	0.30029296874999994	0	0.35034179687499994	1	"2016-07-24 14:53:13.376"	3
1	6.436767578125	0.30029296874999994	0	0.35034179687499994	1	"2016-07-24 14:52:55.396"	3
0	6.436767578125	0.30029296874999994	0	0.35034179687499994	1	"2016-07-24 14:52:41.651"	3
2	6.436767578125	0.30029296874999994	0	0.35034179687499994	0	"2016-07-24 14:52:37.05"	3
3	6.436767578125	0.30029296874999994	0	0.35034179687499994	0	"2016-07-24 14:52:39.684"	3
1	6.25732421875	0.17944335937499994	0.17944335937499994	0.17944335937499994	2	"2016-07-24 14:23:08.071"	3
3	6.436767578125	0.17944335937499994	0.17944335937499994	0.17944335937499994	2	"2016-07-24 14:23:09.255"	3
1	6.436767578125	0.30029296874999994	0	0.35034179687499994	0	"2016-07-24 14:52:34.802"	3
0	6.436767578125	0.30029296874999994	0	0.35034179687499994	0	"2016-07-24 14:52:20.038"	3
0	6.077880859375	0.17944335937499994	0.17944335937499994	0.17944335937499994	2	"2016-07-24 14:23:07.005"	3
1	5.53955078125	0.17944335937499994	0.17944335937499994	0.17944335937499994	1	"2016-07-24 14:23:02.056"	3
3	5.8984375	0.17944335937499994	0.17944335937499994	0.17944335937499994	1	"2016-07-24 14:23:04.974"	3
2	5.718994140625	0.17944335937499994	0.17944335937499994	0.17944335937499994	1	"2016-07-24 14:23:03.856"	3

Figure 5.13: Screenshot of Collection UserResponse Part 1

user_id	improve	privacy_can_be	privacy_gain	privacy_level	privacy_percentage	sensors
"57935b55a67b0ba32f81eeac"	1	1.5625	0.78125	3	63.28125	1
"57935b55a67b0ba32f81eeac"	2	0	0	5	63.28125	2
"57935b55a67b0ba32f81eeac"	2	0	0	5	63.28125	2
"57935b55a67b0ba32f81eeac"	2	0	0	5	63.28125	2
"57935b55a67b0ba32f81eeac"	2	0	0	5	63.28125	2
"57935b55a67b0ba32f81eeac"	2	0	0	5	63.28125	2
"57935b55a67b0ba32f81eeac"	2	0	0	5	63.28125	2
"57935b55a67b0ba32f81eeac"	2	0	0	5	63.28125	2
"57935b55a67b0ba32f81eeac"	2	0	0	5	63.28125	2
"57935b55a67b0ba32f81eeac"	2	0.5504352278545781	-1.036866359…	1	64.28571428571429	1
"57935b55a67b0ba32f81eeac"	2	0.5580357142857082	-1.004464285…	1	63.28125	1
"57935b55a67b0ba32f81eeac"	2	0	0	5	63.28125	2
"57935b55a67b0ba32f81eeac"	2	0	0	5	63.28125	2
"57935b55a67b0ba32f81eeac"	2	0.542041248016929	-1.070861977…	1	65.3225806451613	1
"57935b55a67b0ba32f81eeac"	2	0.5113968439509051	-1.183518410…	1	68.64406779661017	1
"57935b55a67b0ba32f81eeac"	2	0.5327868852458977	-1.106557377…	1	66.39344262295081	1
"57935b55a67b0ba32f81eeac"	2	0.5225988700564983	-1.144067796…	1	67.5	1

Figure 5.14: Screenshot of Collection UserResponse Part 2

2. Script for summarization

The stakeholder make a request for data giving the following details:

1. Bidding day number
2. Anonymous user
3. Sensor

day_no	user_id	lat	long	summarization	timestamp
3	"578e91e778f2511711cfb9f5"	47.419864654541016	8.502890586853027	1	1469186498206
3	"578e91e778f2511711cfb9f5"	47.419864654541016	8.502890586853027	1	1469186468203
3	"578e91e778f2511711cfb9f5"	47.419864654541016	8.502890586853027	1	1469186408196
3	"578e91e778f2511711cfb9f5"	47.419864654541016	8.502890586853027	1	1469186438200
3	"578e91e778f2511711cfb9f5"	47.419864654541016	8.502890586853027	1	1469186378192
3	"578e91e778f2511711cfb9f5"	47.419864654541016	8.502890586853027	1	1469186288183
3	"578e91e778f2511711cfb9f5"	47.419864654541016	8.502890586853027	1	1469186348189
3	"578e91e778f2511711cfb9f5"	47.419864654541016	8.502890586853027	1	1469186258180
3	"578e91e778f2511711cfb9f5"	47.419864654541016	8.502890586853027	1	1469186228177
3	"578e91e778f2511711cfb9f5"	47.419864654541016	8.502890586853027	1	1469186318186
3	"578e91e778f2511711cfb9f5"	47.419864654541016	8.502890586853027	1	1469186198171
3	"578e91e778f2511711cfb9f5"	47.419864654541016	8.502890586853027	1	1469186168166
3	"578e91e778f2511711cfb9f5"	47.419864654541016	8.502890586853027	1	1469186138163
3	"578e91e778f2511711cfb9f5"	47.419864654541016	8.502890586853027	1	1469186108160
3	"578e91e778f2511711cfb9f5"	47.419864654541016	8.502890586853027	1	1469186078154
3	"578e91e778f2511711cfb9f5"	47.419864654541016	8.502890586853027	1	1469186018145
3	"578e91e778f2511711cfb9f5"	47.419864654541016	8.502890586853027	1	1469186048150

Figure 5.15: Screenshot of Collection Location

day_no	summarization	timestamp	user_id	x	y	z
3	1	1469186493918	"578e91e778f251171…	0.0191536135971546…	-0.143652096390724…	10.15141487121582
3	1	1469186463719	"578e91e778f251171…	0.0191536135971546…	-0.143652096390724…	10.15141487121582
3	1	1469186373308	"578e91e778f251171…	0.0191536135971546…	-0.1244984865188594…	10.180145263671875
3	1	1469186433709	"578e91e778f251171…	0.0287304203957319…	-0.134075298905372…	10.15141487121582
3	1	1469186343108	"578e91e778f251171…	0.0287304203957319…	-0.134075298905372…	10.15141487121582
3	1	1469186403508	"578e91e778f251171…	0.0191536135971546…	-0.134075298905372…	10.15141487121582
3	1	1469186282709	"578e91e778f251171…	0.0191536135971546…	-0.143652096390724…	10.15141487121582
3	1	1469186222308	"578e91e778f251171…	0.0287304203957319…	-0.1053448766469955…	10.20887565612793
3	1	1469186252509	"578e91e778f251171…	0.0287304203957319…	-0.143652096390724…	10.160991668701172
3	1	1469186312909	"578e91e778f251171…	0.0287304203957319…	-0.143652096390724…	10.15141487121582
3	1	1469186131909	"578e91e778f251171…	0.0191536135971546…	-0.143652096390724…	10.160991668701172
3	1	1469186192307	"578e91e778f251171…	0.0287304203957319…	-0.134075298905372…	10.160991668701172
3	1	1469186162108	"578e91e778f251171…	0.0191536135971546…	-0.143652096390724…	10.160991668701172
3	1	1469186101709	"578e91e778f251171…	0.0287304203957319…	-0.143652096390724…	10.15141487121582
3	1	1469186011488	"578e91e778f251171…	0.0191536135971546…	-0.143652096390724…	10.132261276245117
3	1	1469185981478	"578e91e778f251171…	0.0287304203957319…	-0.143652096390724…	10.160991668701172
3	1	1469186071708	"578e91e778f251171…	0.0191536135971546…	-0.143652096390724…	10.15141487121582

Figure 5.16: Screenshot of Collection Accelerometer

4. Context

Given this input plus the category of the stakeholder, we look into the User-Response Collection trying to find the most recent record that fit this criteria and extract the records privacy level.

Once we know the privacy level, summarization can be done. Data has been taken from the user with a certain summarization, and if the summarization

bands	user_id	day_no	rms	spl	summarization	timestamp
"0.0,1.9080862E-5,…	"578e91e778f2511711cfb9f5"	3	107.31494140625	62.65440368652344	1	1469186468205
"0.0,1.5665331E-5,…	"578e91e778f2511711cfb9f5"	3	88.882568359375	61.01753234863281	1	1469186498214
"0.0,1.952634E-5,5…	"578e91e778f2511711cfb9f5"	3	100.1298828125	62.05247497558594	1	1469186438208
"0.0,2.1573624E-5,…	"578e91e778f2511711cfb9f5"	3	47.744140625	55.61960220336914	1	1469186408210
"0.0,2.0647063E-5,…	"578e91e778f2511711cfb9f5"	3	73.727294921875	59.39376449584961	1	1469186378210
"0.0,2.1016425E-5,…	"578e91e778f2511711cfb9f5"	3	71.015380859375	59.0682487487793	1	1469186348202
"0.0,2.062715E-5,6…	"578e91e778f2511711cfb9f5"	3	96.7724609375	61.7562370300293	1	1469186258193
"0.0,2.1374477E-5,…	"578e91e778f2511711cfb9f5"	3	67.723876953125	58.656036376953125	1	1469186168186
"0.0,1.7658736E-5,…	"578e91e778f2511711cfb9f5"	3	106.538818359375	62.59135818481445	1	1469186198197
"0.0,1.8755187E-5,…	"578e91e778f2511711cfb9f5"	3	40.89111328125	54.27377700805664	1	1469186318192
"0.0,1.429928E-5,4…	"578e91e778f2511711cfb9f5"	3	21.7080078125	48.773597717285156	1	1469186078184
"0.0,1.9742341E-5,…	"578e91e778f2511711cfb9f5"	3	85.58349609375	60.68899917602539	1	1469186138189
"0.0,2.134739E-5,5…	"578e91e778f2511711cfb9f5"	3	78.437744140625	59.93170166015625	1	1469186108185
"0.0,2.007436E-5,5…	"578e91e778f2511711cfb9f5"	3	82.403076171875	60.360069274902344	1	1469186048176
"0.0,2.0650641E-5,…	"578e91e778f2511711cfb9f5"	3	76.4638671875	59.710323333740234	1	1469185958181
"0.0,1.4806586E-5,…	"578e91e778f2511711cfb9f5"	3	21.33837890625	48.624427795410156	1	1469186018178
"0.0,2.0179677E-5,…	"578e91e778f2511711cfb9f5"	3	101.021484375	62.12947463989258	1	1469185988177

Figure 5.17: Screenshot of Collection Noise

day_no	summarization	timestamp	user_id	x
3	3	1469447239362	"57935b55a67b0ba32…	47
3	3	1469447109437	"57935b55a67b0ba32…	54
3	3	1469447319071	"57935b55a67b0ba32…	39
3	3	1469446323286	"57935b55a67b0ba32…	109
3	3	1469446998112	"57935b55a67b0ba32…	180
3	3	1469446812605	"57935b55a67b0ba32…	165
3	3	1469446228120	"57935b55a67b0ba32…	83
3	3	1469445977205	"57935b55a67b0ba32…	96
3	3	1469445805362	"57935b55a67b0ba32…	156
3	3	1469445621109	"57935b55a67b0ba32…	157
3	3	1469445343373	"57935b55a67b0ba32…	136
3	3	1469445541953	"57935b55a67b0ba32…	143
3	3	1469445255903	"57935b55a67b0ba32…	150
3	3	1469444855996	"57935b55a67b0ba32…	127
3	3	1469444963549	"57935b55a67b0ba32…	127
3	3	1469445171668	"57935b55a67b0ba32…	100

Figure 5.18: Screenshot of Collection Light

level is lower than the privacy level extracted, further summarization needs to be done. The pseudocode is shown in 6.

user_id
"57a8f8f1848532cf76b0836f"
"579a148f352257bc0612c70b"
"57975541e813f99735dd0598"
"57975159890927b613d0f49f"
"57935b55a67b0ba32f81eeac"
"579357faa67b0ba32f81e724"
"57931cad866a46bd554a1896"
"57930d55837af5db6734a438"
"5792471c493006891e4e0c2b"
"57923fbfc3d7cee306b50957"
"57923946c3d7cee306b4fb44"
"5792373d3692318e3ba9e929"
"5792373d3692318e3ba9e929"
"57922e1afb55917415770d44"
"57922a329bb316492f70242b"
"579224ffba46365901e66e78"
"578e91e778f2511711cfb9f5"

Figure 5.19: Screenshot of Collection Users

timestamp	user_id	credit	day_no	privacy
"2016-08-08 23:35:20.788"	"57a8f8f1848532cf76b0836f"	6.310096153846153	1	74.609375
"2016-07-28 16:23:21.687"	"579a148f352257bc0612c70b"	9.030898876404493	1	56.25
"2016-07-26 14:22:12.236"	"57975541e813f99735dd0598"	9.01900773195876	1	57.03125
"2016-07-26 14:09:18.367"	"57975159890927b613d0f49f"	8.850940265486726	1	58.203125
"2016-07-26 14:00:00.565"	"57935b55a67b0ba32f81eeac"	0	4	0
"2016-07-25 13:59:00.084"	"57935b55a67b0ba32f81eeac"	6.510009765625	3	63.28125
"2016-07-24 13:59:31.599"	"57935b55a67b0ba32f81eeac"	8.8885498046875	2	50.390625
"2016-07-23 13:59:31.382"	"57935b55a67b0ba32f81eeac"	5.90576171875	1	67.1875
"2016-07-23 09:31:59.106"	"57931cad866a46bd554a1896"	1.2409156976744176	1	55
"2016-07-23 08:36:40.373"	"57930d55837af5db6734a438"	8.348214285714286	1	60.9375
"2016-07-22 18:18:33.16"	"5792471c493006891e4e0c2b"	1.2428977272727268	1	52.5
"2016-07-22 17:48:06.151"	"57923fbfc3d7cee306b50957"	1.253551136363636	1	55
"2016-07-22 17:19:40.016"	"57923946c3d7cee306b4fb44"	9.385190217391308	1	53.125
"2016-07-22 16:32:46.545"	"57922e1afb55917415770d44"	7.999999999999998	1	63.28125
"2016-07-22 16:22:07.321"	"57922a329bb316492f70242b"	8.459821428571429	1	60.9375
"2016-07-22 13:22:00.152"	"578e91e778f2511711cfb9f5"	14.11313657407408	3	18.359375

Figure 5.20: Screenshot of Collection Score

5.3.2 FairDataShare Web Portal

The FairDataShare portal makes use of a server at ETH Zurich other Kinvey to safely store the usernames, passwords of users and the stakeholders in a collection. The database technology used is MongoDB. The language used to interact with Kinvey is Express.js, which is based on Node.js. Most of the

Algorithm 6 Server Summarization Algorithm

```
1: procedure SUMMARIZATION
2:     data ← sensor data from collection
3:     if summarizationlevel == privacylevel then
4:         Return data
5:     else
6:         skip ← summarizationlevel − privacylevel + 1
7:         for every skipnumber records out of 4 do
8:             Delete record from data
9:     Return data to portal
```

data portal business logic is on Kinvey described in section **??**. The webpage was constructed using Html and css. All screenshots of the portal including detailed information is provided in chapter 4.

Chapter 6

Pre-Survey and Experiment Findings

The following chapter will give an overview of the data obtained from the survey, which was conducted before running the experiment. Later, an overview of the data obtained from the experiment is explained along with feedback received from the participants. This chapter puts forth all that was learnt from the above mentioned.

6.1 Overview of the Pre-Survey Data

The survey has 199 participants. After filtering out spurious and half-filled entries 189 of them are used for the analysis.

6.2 Pre-Survey Methodology and Findings

All the results presented below were performed on the data by performing the following changes to the data:

1. Rows with empty fields were removed
2. Rows with spurious data was removed
3. Data was scaled or normalized when necessary

Other than the above, the data was not manipulated. Outliers were not excluded either.

Perception of Individual Sensor Grouped on the Intrusion of Sensors in General

We try to examine here if the perception of intrusion of Sensors can affect the way a person views the individual sensors themselves. In other words, we try to examine if there is a significant difference in perception of each

sensor depending on the perception of the Sensors as a whole. For this, we grouped the survey data based on the responses to question 10. Since there are 5 possible responses to this question, this makes 5 individual groups from 1 to 5.

We now have 5 groups who view sensors in a different light each and their perception of each of the individual sensors can be compared. Before going into the comparison, we try to understand the properties of the data. To perform a one-way ANOVA test or a t-test, the data needs to be:

1. Normally distributed
2. Homoscedastic
3. Ordinal or continuous

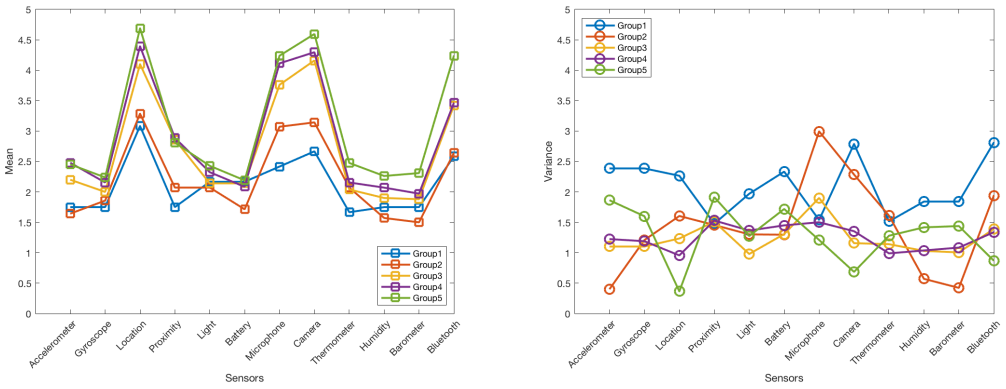
Since the data is discrete and follows the Likert Scale with options from 1 to 5, it gives skewed normal distribution. Additionally, the variances of values within the groups formed are not similar. One-way ANOVA test is quite robust to heteroscedacity, as long as the maximum variance among all groups is less than four times the group with the lowest variance. The scale used to collect data is in the ordinal form. Accounting for all the violations, we instead opt for a non-parametric tests such as the Kruskal-Wallis H test and the Dunn's test which only assume the following :

1. Groups are independant from one another
2. All observations are independant
3. The dependant variables should be in the ordinal scale or continuous

The above tests do not make any assumptions about the distribution of the data and are robust to heteroscedastic data.

Group 1 to 5 have 13, 14, 50, 71 and 42 people each respectively. For in depth analysis of the composition of each group in terms of employment, education, gender and birth year please refer to tables 6.1,6.4,6.2,6.3. Figures 6.1a and 6.1b depict the mean and variances for each of the individual groups. We start by performing the non paramteric Kruskal-Wallis test on each Sensor. The value of alpha assumed here is 0.05. The null hypothesis states that all the groups perceive the sensors in the similar way. This means they come from the same distribution. The alternative hypothesis is that the groups perceive each sensor in a significantly different way. The table 6.5 depicts the p-values obtained from the test.

On these sensors, we proceed with a post hoc test by performing a pariwise Dunn's test to examine if there is an actual significant difference between the groups and if so between which groups. The sensors with p values with less than 0.05 are examined in more detail and the p-values are presented in



(a) Mean of each Group for each Sensor

(b) Variance of each Group for each Sensor

Figure 6.1: Table Schemas

Table 6.1: Employment Classification of Groups

Occupation	1	2	3	4	5
Employed full time	4.90%	6.86%	26.47%	38.24%	23.53%
Employed part time	8.33%	16.67%	33.33%	16.67%	25.00%
Unemployed and looking for work	8.33%	16.67%	16.67%	25.00%	33.33%
Unemployed and not looking for work	0.00%	0.00%	0.00%	66.67%	33.33%
Retired	0.00%	0.00%	100.00%	0.00%	0.00%
Student	7.23%	4.82%	26.51%	39.76%	21.69%
Disabled	0.00%	0.00%	0.00%	0.00%	0.00%

Table 6.2: Gender Classification of Groups

Gender	1	2	3	4	5
Female	5.56%	4.17%	33.33%	30.56%	26.39%
Male	7.14%	9.52%	22.22%	39.68%	21.43%

Table 6.3: Average Birth Year of Groups

1	2	3	4	5
1989	1979	1986	1986	1983

table 6.6. The table shows the results for each pairwise test done, with the p-values adjusted using the Bonferroni Method. The reason for choosing to adjust the p-values is that repeated experiments can increase the chances of

Table 6.4: Education Classification of Groups

Education	1	2	3	4	5
Less than high school	20.00%	20.00%	20.00%	40.00%	0.00%
High school	10.53%	0.00%	52.63%	31.58%	5.26%
Some college	10.00%	20.00%	30.00%	20.00%	20.00%
Bachelors degree	7.02%	10.53%	24.56%	42.11%	15.79%
Masters degree	3.80%	5.06%	24.05%	35.44%	31.65%
PhD degree	7.14%	7.14%	17.86%	35.71%	32.14%

Table 6.5: Kuskal-Wallis Test

Sensor	p-value
Accelerometer	0.0151
Gyroscope	0.2959
Location	1.0664e-05
Proximity	0.0147
Light	0.6933
Battery	0.6950
Microphone	3.0070e-04
Camera	2.1191e-05
Thermometer	0.0693
Air Humidity	0.1292
Barometer	0.0949
Bluetooth	3.4877e-05

accepting the alternative hypothesis so p-values are adjusted according to the number of experiments performed. 10 experiments are performed per sensor.

For the Accelerometer and Proximity, we can see that none of the pairwise groups have a significant difference from each other. This means that even tough the groups perceive sensors differently in general, they all view Accelerometers and Proximity in a similar way.

For the location sensor, it can be observed that groups (1,4), (1,5), (2,4), (2,5) have a significant difference. This can be attributed to the fact that since the groups are formed from the perception of people of the Sensors Feature, the difference in perception between group 1 and group 5 will be larger than between group 1, group 2 and group 1, group 3 since they are not much apart in the scale.

For the microphone sensor, it can be seen that groups (1,3), (1,4) and (1,5) are significantly different from each other. This goes to show that if people

Table 6.6: Dunn's Test 1

Groups	Accelerometer	Location	Proximity	Microphone	Camera	Bluetooh
(1,2)	1.0000	1.0000	0.9992	0.8365	1.0000	1.0000
(1,3)	0.4207	0.2084	0.0699	0.0365	0.0732	0.7825
(1,4)	0.0595	0.0125	0.0513	0.0012	0.0048	0.6442
(1,5)	0.2054	0.0010	0.1191	0.0009	0.0007	0.0029
(2,3)	0.6548	0.1774	0.3713	0.8927	0.1694	0.5921
(2,4)	0.1185	0.0077	0.2617	0.2287	0.0123	0.4270
(2,5)	0.3659	0.0005	0.5184	0.1597	0.0018	0.0007
(3,4)	0.8989	0.7642	1.0000	0.8052	0.9040	1.0000
(3,5)	0.9997	0.0869	1.0000	0.6390	0.2947	0.0066
(4,5)	0.9998	0.8360	1.0000	1.0000	0.9617	0.0059

rate sensors as even a little intrusive, they all rate the microphone's in a significantly different way than the people who rate sensors as non-intrusive.

For the camera sensor, it can be observed that groups (1,4), (1,5), (2,4), (2,5) have a significant difference in their perception of the intrusion. Similar to the location sensor, people with perception of sensors in general with a lower intrusion level have significantly different responses to the camera intrusion than the people who rate sensors with more intrusion.

For the Bluetooth sensor, there is a significant difference between groups (1,5), (2,5), (3,5) and (4,5). This shows that responses by people who find sensors extremely intrusive is different from the rest of the groups.

Perception of Individual Stakeholders Grouped on the Intrusion of Stakeholders in General

In this section, we try to see if the intrusion level perception by people of Stakeholders in general and the intrusion of the individual stakeholders are related. We try to examine significant differences between groups formed by using question 12's responses on the perception of each stakeholder, and since there are 5 different responses this give five independent groups. The groups 1 to 5 have 7, 11, 32, 69 and 70 people in each respectively. More detailed information about the employment, education, gender and age distribution in the groups is given in tables 6.7, 6.10, 6.8 and 6.9.

The mean and variances of each group formed is depicted in figures 6.2a and 6.2b. To start with we examine all the groups simultaneously for all the stakeholder's, we perform the Kruskal-Wallis H test since the data is discrete and not normally distributed. The null hypothesis is that the groups rate the intrusion of a particular stakeholder in a similar way. The alternative

hypothesis is that the groups rate the intrusion of a particular stakeholder in a significantly different way.

The resulting p-values of this test is displayed in table **??**. As it can be seen, the test pronounces that all the groups are significantly different at an alpha with 0.05.

Table 6.7: Employment Classification of Groups

Occupation	1	2	3	4	5
Employed full time	3.92%	4.90%	16.67%	33.33%	41.18%
Employed part time	16.67%	16.67%	8.33%	50.00%	8.33%
Unemployed, looking for work	8.33%	8.33%	16.67%	16.67%	50.00%
Unemployed, not looking for work	0.00%	0.00%	0.00%	33.33%	66.67%
Retired	0.00%	0.00%	0.00%	100.00%	0.00%
Student	4.82%	7.23%	15.66%	37.35%	34.94%
Disabled	0.00%	0.00%	0.00%	0.00%	0.00%

Table 6.8: Gender Classification of Groups

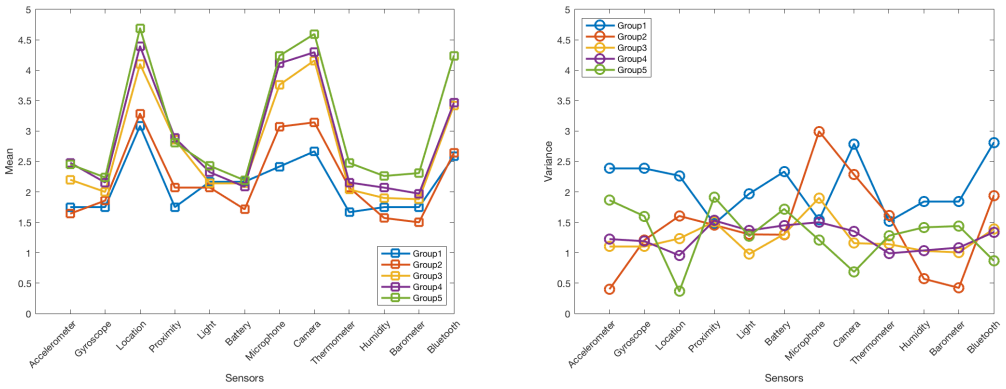
Gender	1	2	3	4	5
Female	5.56%	6.94%	23.61%	36.11%	27.78%
Male	3.97%	6.35%	11.90%	35.71%	42.06%

Table 6.9: Average Birth Year of Groups

1	2	3	4	5
1986	1989	1984	1985	1984

Table 6.10: Education Classification of Groups

Education	1	2	3	4	5
Less than high school	20.00%	20.00%	0.00%	60.00%	0.00%
High school	15.79%	5.26%	21.05%	31.58%	26.32%
Some college	0.00%	10.00%	20.00%	40.00%	30.00%
Bachelors degree	1.75%	12.28%	15.79%	35.09%	35.09%
Masters degree	5.06%	1.27%	13.92%	37.97%	41.77%
PhD degree	0.00%	7.14%	21.43%	28.57%	42.86%



(a) Mean of each Group for each Sensor

(b) Variance of each Group for each Sensor

Figure 6.2: Table Schemas

Table 6.11: Kuskal-Wallis Test

Stakeholder	p-value
Corporation	2.1432e-05
Non-Governmental Organization	0.0221
Educational Institution	0.0396
Government	0.0024

This prompts us to take a closer look at which of the groups are significantly different from each other for each stakeholder. For this we continue the experiment with Dunn's Test with p-values adjusted by the Bonferroni Method. The results from the test is shown in table 6.12.

Table 6.12: Dunn's Test 1

Groups	Corporation	Non-Governmental Organization	Educational Institution	Government
(1,2)	0.9839	0.8042	0.9565	0.6615
(1,3)	0.3282	0.2351	0.8540	0.0986
(1,4)	0.0240	0.1962	0.6867	0.0289
(1,5)	0.0012	0.0243	0.1254	0.0028
(2,3)	0.9467	0.9992	1.0000	0.9958
(2,4)	0.2047	0.9991	1.0000	0.9252
(2,5)	0.0110	0.7192	0.8415	0.3670
(3,4)	0.6893	1.0000	1.0000	0.9999
(3,5)	0.0197	0.8906	0.4112	0.5794
(4,5)	0.4640	0.6027	0.3427	0.7378

The test was done for all stakeholders since the Kruskal-Wallis test denoted that all groups differ significantly for all stakeholders. Looking at the stakeholder corporation, we see that groups (1,4), (1,5) and (2,5). This goes to show that groups with larger difference in their outlook to stakeholders as a whole view corporations in a significantly different way.

For the Non-Governmental Organization, only the groups (1,5) differ significantly. This goes to show that groups that do not find stakeholders intrusive and groups that find stakeholders very intrusive rate the intrusion of Non-Governmental Organization in significantly different ways.

For Educational Institutions, none of pairwise comparisons have p-values below 0.05. This goes to show that the intrusion of Educational Institutions by all groups does not differ significantly.

Lastly, for the stakeholder Government, the groups (1,4) and (1,5) differ significantly. This goes to show that groups that view the intrusion of stakeholders with a larger difference view Government in a significantly different way.

The trend observed above is that there is a significant difference in the outlook of individual stakeholders between groups with larger differences in their outlook to stakeholders as a whole, with the exception of Educational Institution where the alternative hypothesis was rejected.

Perception of Individual Contexts Grouped on the Intrusion of Contexts in General

In this section, we examine the examine the relationship between the intrusion of Contexts in General and the individual contexts in question 13. To do this, like the above sections we partition the data into groups based on the answers given questions 14, which asks the user the perception of intrusion of Contexts in general. There are five groups in total. Group one to five have each 12, 11, 42, 74 and 50 people respectively. Additional information about the groups on enployment, education , gender and year of birth is given in tables 6.13, 6.16, 6.14 and 6.15.

Like in the previous cases, since the data is discrete and not normal, we use the Kruskal-Wallis test to compare the groups perceptions on various contexts. The alpha value is considered to be 0.05. The results are presented in table 6.17. As it can be seen, the test says that there is a significant difference between the groups for all sensors.

We not perform Dunn's Test as a post hoc test for all the contexts to observe the exact group pairs that might be significantly different. The results are presented in tables 6.18 and 6.19. For the context Education, the groups (2,5) and (3,5) are significantly different from each other. For the context Entertainment, further investigation shows that groups (1,5), (2,5) and (3,5) are

Table 6.13: Employment Classification of Groups

Occupation	1	2	3	4	5
Employed full time	4.90%	3.92%	23.53%	36.27%	31.37%
Employed part time	16.67%	8.33%	25.00%	25.00%	25.00%
Unemployed, looking for work	8.33%	16.67%	25.00%	25.00%	25.00%
Unemployed, not looking for work	0.00%	0.00%	0.00%	66.67%	33.33%
Retired	0.00%	0.00%	0.00%	100.00%	0.00%
Student	7.23%	6.02%	20.48%	44.58%	21.69%
Disabled	0.00%	0.00%	0.00%	0.00%	0.00%

Table 6.14: Gender Classification of Groups

Gender	1	2	3	4	5
Male	7.14%	6.35%	23.02%	38.10%	25.40%
Female	5.56%	5.56%	19.44%	38.89%	30.56%

Table 6.15: Average Birth Year of Groups

1	2	3	4	5
1986	1986	1986	1985	1983

Table 6.16: Education Classification of Groups

Education	1	2	3	4	5
Less than high school	20.00%	0.00%	0.00%	80.00%	0.00%
High school	10.53%	10.53%	31.58%	36.84%	10.53%
Some college 1	0.00%	0.00%	40.00%	30.00%	20.00%
Bachelors degree	7.02%	8.77%	24.56%	35.09%	24.56%
Masters degree	6.33%	3.80%	17.72%	40.51%	31.65%
PhD degree	0.00%	7.14%	17.86%	35.71% 3	9.29%

significantly different in each others responses. For the context Environment, groups (2,5) and (3,5) are significantly different from each other. For the context Finance, the groups (1,5), (3,5) and (4,5) are significantly different from each other. In the context health, except for groups (1,5) all the other groups are not significantly different from each other. For the context Shopping, the groups (1,5), (3,4) and (3,5) are significantly different from each other. For the context Social Network, none of the groups are significantly different from each other. In the context Training, the groups (1,5),(3,5) and (4,5) are significantly different from each other. Finally for the context Transportation,

Table 6.17: Kuskal-Wallis Test

Context	p-value
Education	6.4694e-04
Entertainment	1.0660e-04
Environment	1.3079e-04
Finance	0.0021
Health	0.0011
Shopping	5.4227e-05
Social Network	0.0120
Training	1.2071e-05
Transportation	4.9043e-04

Table 6.18: Dunn's Test Part 1

Groups	Education	Entertainment	Environment	Finance	Health
(1,2)	0.99796	0.99982	0.97055	1.0000	0.63908
(1,3)	1.0000	0.99174	1	0.32963	0.076147
(1,4)	0.94148	0.19262	0.86998	0.22077	0.042547
(1,5)	0.11471	0.0056283	0.21553	0.015364	0.00051115
(2,3)	0.9342	1	0.96665	0.31604	0.9999
(2,4)	0.32744	0.76121	0.083389	0.21508	0.99963
(2,5)	0.0082212	0.082023	0.0048936	0.016365	0.50246
(3,4)	0.83617	0.23034	0.10875	1.0000	1.0000
(3,5)	0.0064191	0.00086418	0.0012905	0.65747	0.32713
(4,5)	0.13825	0.28231	0.60016	0.57519	0.21258

Table 6.19: Dunn's Test Part 2

Groups	Shopping	Social Network	Training	Transportation
(1,2)	1.0000	0.99831	1.0000	1.0000
(1,3)	0.99992	0.94767	1.0000	0.9722
(1,4)	0.18309	0.089135	0.90197	0.23809
(1,5)	0.015842	0.10636	0.010906	0.016376
(2,3)	1.0000	1.0000	1	0.98253
(2,4)	0.39426	0.70552	0.99778	0.30509
(2,5)	0.052334	0.7272	0.068368	0.026144
(3,4)	0.038351	0.21259	0.58123	0.51397
(3,5)	0.00050454	0.29374	2.1697e-05	0.012924
(4,5)	0.69535	1.0000	0.0033189	0.55629

the groups (1,5), (2,5) and (3,5) are significantly different from each other.

This goes to show that groups that perceive contexts in a more different light tend to have different ways of viewing the individual contexts. We do observe that in some cases (2,5) are significantly different, but (1,5) is not. This can be easily attributed to the low number of responses and the noise in the data.

K- Means to Automatically detect clusters in the Population

We attempt to cluster the population that answered the survey based on the following traits :

- The way people rate data collectors (question 11)
- The way people rate sensors (question 9)
-

For each chosen group of features, we need to discover how many clusters there are in this population. Since we do not have any pre-defined labels given to us, we go through the following procedure:

- Perform the K-Means clustering using various distances
  - Squared Euclidean Distance
  - City Block Distance
  - Cosine Distance
- We plot the Silhouette score to see which of the distances do better
- Visualize the clusters in the highest principal components

6.3 Overview of the Experiment Data

6.4 Findings from the Experiment Data

Chapter 7

Conclusion

Appendix A

Appendix

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.