Real-time Monitoring of a Rich Social Data Infrastructure

Zhi Zhang

Kongens Lyngby 2013
IMM-M.Sc.-2013-58

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk IMM-M.Sc.-2013-58

Abstract

In recent years, studies find out that only big dataset is not enough to advance the quantitative understanding of social systems, high quality data is also needed. At the Technical University of Denmark, Sensible DTU project aims to gain knowledge about social systems by creating a dataset of unparalleled quality and size based on the capabilities of modern smartphones. Due to the data plays the most important role in the project, ensuring the user's phone collecting data properly is a crucial work.

This thesis is a case study of designing and implementing a data reporting system in real time that gives the user an easy approach for checking the data quality in the database. The information of data quality is visualized from different aspects with different level of details. To present as much information as possible and improve the user experience, a variety of visualizations and interaction techniques are used.

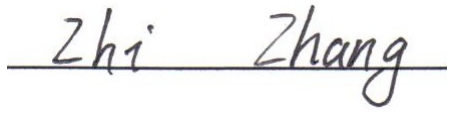
Before the implementation of the system, a prototype is implemented. With the feedback from the prototype, a web application called Sensible DTU Data Monitor is developed, which is the outcome of the thesis. An evaluation consisting of user feedback and performance tests was conducted. Based on the evaluation, there is a discussion of the thesis result and future works.

Preface

This thesis was prepared at the department of Informatics and Mathematical Modelling at the Technical University of Denmark in fulfilment of the requirements for acquiring the degree of Master of Science in Engineering (Computer Science and Engineering).

The thesis supervisors are Sune Lehmann Jørgensen and Jakob Eg Larsen, Department of Informatics and Mathematical Modelling, Technical University of Denmark.

Lyngby, 19-July-2013



Zhi Zhang

Acknowledgements

First of all, I am grateful for the opportunity to have Sune Lehmann Jørgensen and Jakob Eg Larsen as my advisors. You have helped me a lot with my work. I cannot finish my thesis without your support and valuable advices. I have a really happy time working on the Sensible DTU project.

Furthermore, I would like to thank all the people working on Sensible DTU. I have learned a lot of interesting stuff and got inspirations on my own work during the group meetings.

Finally, I would like to thank my friends who helped with testing of the system. You have given me a lot of useful feedback and advices for improving the system and visualizations.

Thank you.

Contents

Chapter 1

Introduction

1.1 Sensible DTU

In recent years, digital traces of our daily lives are being recorded in great detail, which holds the potential to fundamentally change how we quantify and understand human behavior and human nature [1]. Furthermore, only big dataset is not enough to advance our quantitative understanding of social systems, we also need high quality data. At the Technical University of Denmark, a research project named Sensing high resolution complex networks is established for creating a dataset of unparalleled quality and size. The high quality data is based on collecting high resolution data from multi-communication channels, in which the high resolution is high frequency of sampling.

The public facing side of the project is called Sensible DTU [2]. A team of researchers from DTU in collaboration with researchers from sociology, anthropology and other social sciences decided to investigate via smartphones with specially designed equipment. The project has two basic goals. The first goal is to get a deeper understanding of the social networks, which can help to solve important societal problems that prevention of epidemics or prevention of social inequality. The other goal is making student life better, which is to give tomorrow's technology in the hands of DTU students and a smartphone who knows your social network could be things like no other phone can.

In current, the project has established a highly scalable computational backend infrastructure and developed a robust and reliable app for data collection. The participants of Sensible DTU project were DTU students selected from four fields of study based on their ideas and thoughts about the project. In these programs of study, all participants got the opportunity to borrow a smartphone (Samsung Galaxy Nexus) in a year so they can participate in the trial. The data collector on the smartphone keeps collecting data from the user, which includes data from Bluetooth, GPS, Wi-Fi, Cell Tower, contact, SMS, call log and hardware status.

In the project, all of the researches are based on the data collected from users so that the data plays the most important role. As the consequent, ensuring the user's phone collecting data properly is a crucial work in the project. Currently, there are 135 users participated in Sensible DTU project and the number of user will increase to more than one thousand soon. Finding an efficient way to monitor a big amount of users' data quality becomes more and more important.

1.2 Goals

The goal of this thesis is to design and implement a data reporting system in real time. The system should be able to classify the quality of data collected from the user. The information about data collecting will be visualized from different aspects with different level of details. The top level is the overview of system working status with statistic data. On the second level, the data quality of individual user is presented. The detail information about data from each sensor will be the bottom level. The outcome is a working web application to visualize the data qualities in the Sensible DTU project.

1.3 Data Quality

In recent years, data quality has gained more and more importance in theory and practice due to an extended use of data warehouse systems and management support systems. In general, data quality can correspond to any form of data completeness, accuracy and consistency, or any combination of these. The definitions of these data dimensions are as following [3]:

Completeness The extent to which data is not missing and is of sufficient breadth and depth for the task at hand. It is very common that values

for some fields of the data set are missing.

Accuracy The extent to which data is correct and reliable. Errors can be introduced during data collecting.

Consistency The extent to which data is presented in the same format. The whole data set may be not consistent in terms of data types.

Studies have confirmed data quality is a multi-dimensional concept [3, 4]. It may include several concepts, including accuracy, reliability, timeliness, relevance, completeness, consistency and certainty. There is no consensus or universally recognized definition for data quality. Several aspects of data quality are discussed below.

1.3.1 Completeness

The presence of missing data is a ubiquitous problem in data collection. In a typical data set, data may be missing for some fields for some records. The cause of missing data may have different sources such as equipment malfunctions, absence of participants, respondents refusing to answer certain questions and so on. In surveys, for example, people often overlook or forget to answer some of the questions or just do not know the answer.

In data analysis, several methods have been proposed in the literature to treat missing data. In general, missing data treatment methods can be divided into the following three categories [5, 6]:

Ignoring and discarding data There are two main ways to discard data with missing values. The first one is known as complete case analysis. It is available in all statistical packages and is the default method in many programs. This method consists of discarding all instances with missing data. The second method is known as discarding instances and/or attributes. This method consists of determining the extent of missing data on each instance and attribute, and deletes the instances and/or attributes with high levels of missing data. Before deleting any attribute, it is necessary to evaluate its relevance to the analysis. Unfortunately, relevant attributes should be kept even with a high degree of missing values.

Parameter estimation Maximum likelihood procedures are used to estimate the parameters of a model defined for the complete data.

Imputation Imputation is a class of procedures that aims to fill in the missing values with estimated ones. The objective is to employ known relationships that can be identified in the valid values of the data set to assist in estimating the missing values.

1.3.2 Uncertainty

Uncertainty is another aspect of data quality. It includes statistical variations or spread, errors and differences, minimum-maximum range values, and noise. Uncertainty can be classified into these categories [7]:

Statistical Either given by the estimated mean and standard deviation, which can be used to calculate a confidence interval, or an actual distribution of the data.

Error A difference, or an absolute valued error among estimates of the data, or between a known correct datum and an estimate.

Range An interval in which the data must exist, but which cannot be quantified into either the statistical or error definition.

The major source of uncertainty of data is from data acquisition. It is clear that all data sets, whether from instrument measurements or numerical models, have a statistical variation. Another source of data uncertainty is from data transformation. Raw data are sometimes not rendered directly but are subject to further transformations with or without the knowledge of the person doing the visualization task.

1.4 Information Visualization

Information visualization is a rapidly expanding area of research, which is due to dramatic increase in both the size and the number of datasets that need to be visualized. Visualization is an increasingly important technique for the exploration and analysis of the large, complex data sets. While algorithmic analysis can be used to quickly and accurately process data to identify patterns and outliers, it is dependent on having a computational model of the phenomena of interest. The problem is that one may not know what one is looking for, or may not be able to set fixed parameters and thresholds to effectively guide the analysis. Visualization, on the other hand, uses the human perceptual system

to extract meaning from the data, focus attention, and reveal structure and patterns. [8]

Information visualization is an external representation of data that exploits human visual processing to reduce the cognitive loads of task. Endeavors that require understanding global or local graph structure can be handled more easily when that structure is interpreted by the visual processing centers of the brain, often without conscious attention, than when that structure has to be cognitively inferred and kept in working memory. External representations change the nature of a task: an external memory aid anchors and structures cognitive behavior by providing information that can be directly perceived and used without being interpreted and formulated explicitly. [9]

The principle of information visualization can be recapped as [10]:

Visual encoding In all visualization, graphical elements are used as a visual syntax to represent semantic meaning. For instance, color can be used to represent the temperature of a place in a weather map where red represents hot and white or blue represents cold, even though the blue color has the highest color temperature. We call these mappings of information to display elements visual encodings, and the combination of several encodings in a single display results in a complete visual metaphor.

Interactions Interactivity is the great challenge and opportunity of information visualization. The advent of computers sets the stage for designing interactive visualization systems of unprecedented power and flexibility.

Chapter 2

Related Work

2.1 Visualization

In an age where we have access to more information than ever before, visualization has become a popular method for making enormous sets of data intelligible. Visualizing data interactively by using different dynamic presentations that rely on graphs, charts, maps and other techniques is often a powerful way to make sense from a vast amount of gathered data. There are many studies covers the topic of information visualization. For example, book "Visualizing Data" [11] introduces the process of creating good visualizations. While the world of visualization is not simple enough to construct a do-and-do-not-list, special considerations must be taken. "Beautiful data" [12] and "Beautiful visualization" [13] tell more stories behind the visualization.

2.2 Data Quality Visualization

Data quality visualization issues have been studied by many different research communities. In paper "Exploratory visualization of multivariate data with variable quality" [14], Xie Z and Huang S provide some techniques for data

uncertainty visualization. Firstly, they define a data structure for quality measures. As shown in Figure 2.1, the quality measures consist of a vector of values for the record quality (one entry per record), a vector for the dimension quality (one entry per dimension), and a two dimensional table of values for the data value quality (one entry per value in the original dataset).

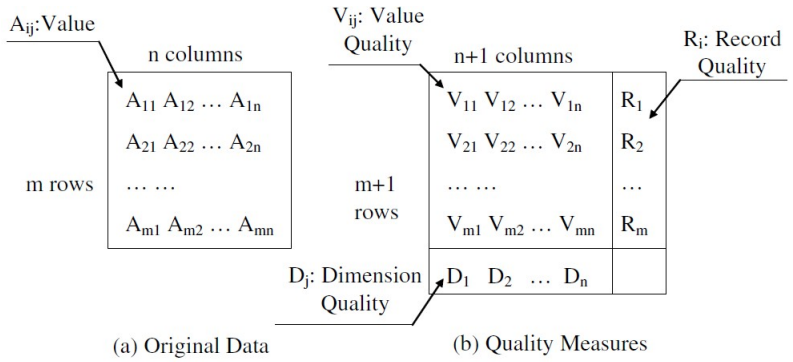


Figure 2.1: The structure of data quality [14]

One approach for uncertainty visualization is embodying data value quality and record quality into the original dataset as new dimensions, then, the quality-extended dataset can be visualized directly using multivariate data visualizations. Figure 2.2 shows an example of visualization of quality-extended dataset using parallel coordinates. Each value quality axis is put beside the corresponding data dimension axis and the last axis is record quality.

Another approach is integrating quality attributes in data visualizations as visual variables, such as size, color and position. The selection of visual variables is one of the key factors in determining whether the visualization can enable users to interpret the quality information and draw reliable conclusions quickly. Some of the visual variables can be used for conveying either data values or quality attributes, for example, size (length, width), blur, color (hue, saturation, brightness) and position (2D, 3D). The selection of visual variables for integrating quality attributes depends on the context of the visualization. Figure 2.3 shows a visualization using parallel coordinates, in which the value quality encoded as line width, record quality as hue and dimension quality as line width.

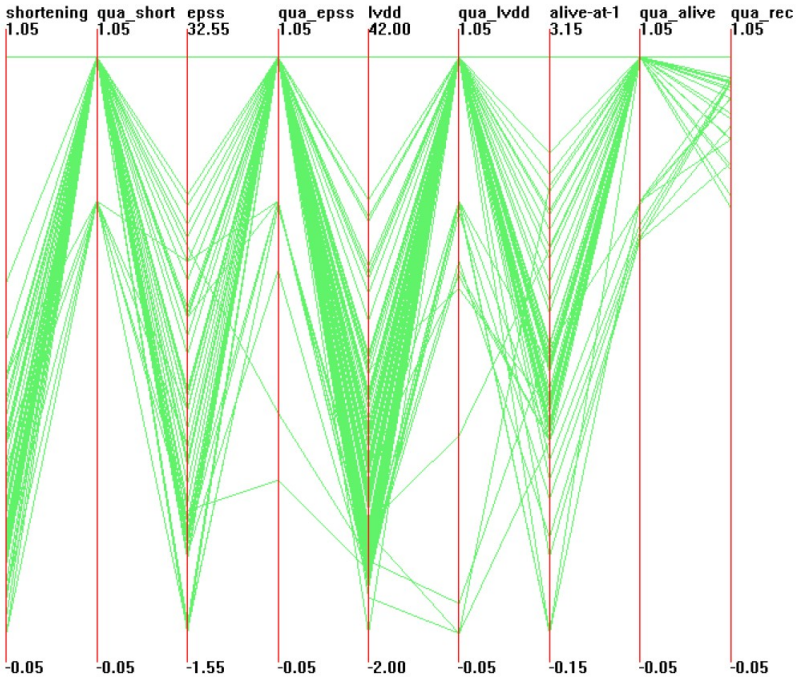


Figure 2.2: Enlarged dataset with quality measures visualized using parallel coordinates [14]

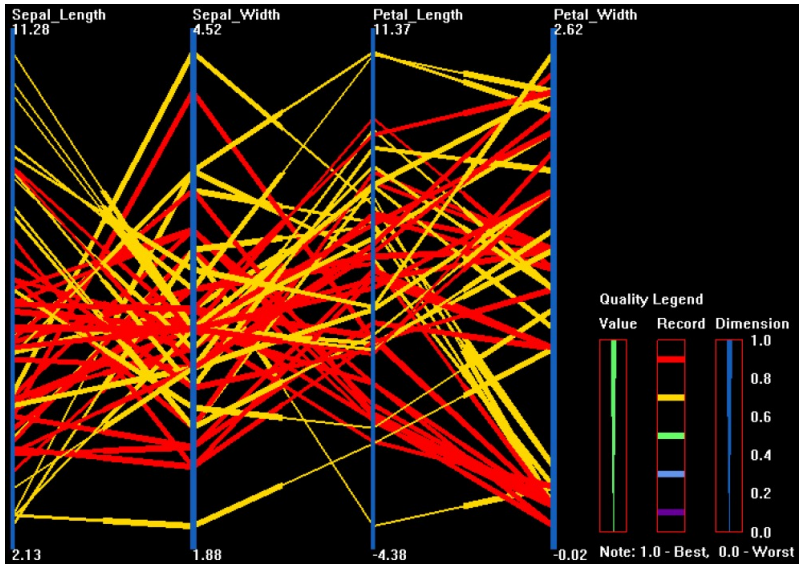


Figure 2.3: Integrating quality attributes in visual variables [14]

2.3 Visualization Interaction

Interaction is an essential part of information visualization. Without interaction, an information visualization technique or system becomes a static image or autonomously animated images. While static images clearly have analytic and expressive value, their usefulness becomes more limited as the data set that they represent grows larger with more variables.

The work of "Toward a deeper understanding of the role of interaction in information visualization" [15] is to identify the fundamental ways that interaction is used in information visualization systems and the benefits it provides to them. They propose seven general categories of interaction techniques widely used in information visualizations:

2.3.1 Select: mark something as interesting

Select interaction techniques provide users with the ability to mark data items of interest to keep track of it. When too many data items are presented on a view, or when representations are changed, it is difficult for users to follow items of interest. As shown in Figure 2.4, Dust & Magnet visualizes data items

as specks of iron that move when magnets (attributes) are manipulated. The selected items are labeled in red, so even after rearranging items users can easily track and identify the location of items of interest.

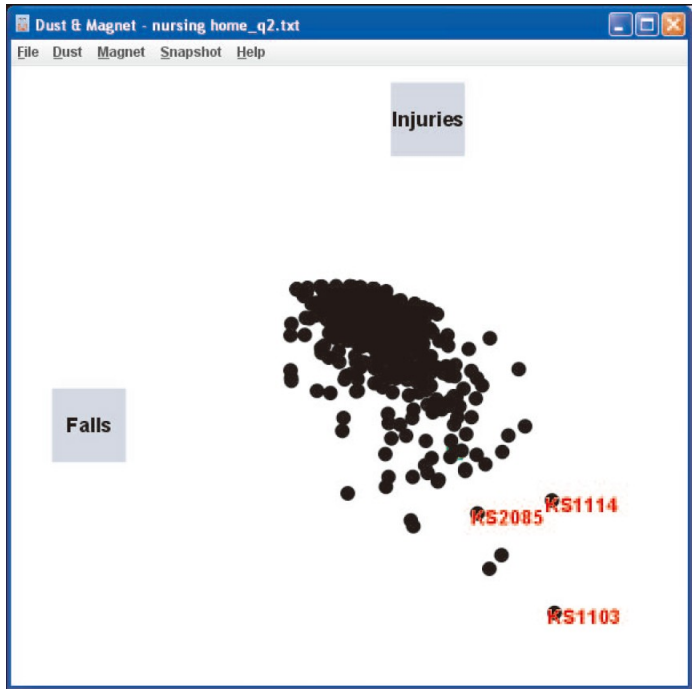


Figure 2.4: A screen shot of Dust & Magnet showing the marking feature. [15]

2.3.2 Explore: show me something else

Explore interaction techniques enable users to examine a different subset of data cases. When users view large scale data set using an information visualization system, they often can only see a limited number of data items at a time. Users typically examine a subset of the data to gain understanding and insight, and then they move on to view some other data. The most common Explore interaction technique is panning. Panning refers to the movement of a camera across a scene or scene movement while the camera stays still.

2.3.3 Reconfigure: show me a different arrangement

Reconfigure interaction techniques provide users with different perspectives onto the data set by changing the spatial arrangement of representations. One of the essential purposes of information visualization is to reveal hidden characteristics of data and the relationships between them. Reconfigure interaction techniques allow users to change the way data items are arranged or the alignment of data items in order to provide more perspectives than static representation. As shown in Figure 2.5, by sorting the "Horsepower" column, users can determine that horsepower values of vehicles are roughly correlated with cylinders, displacement, and weight.

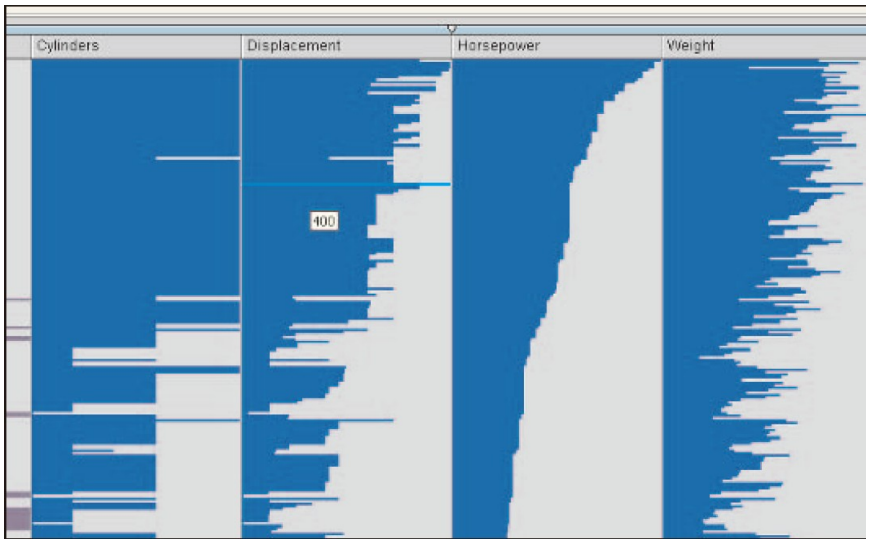


Figure 2.5: A screen shot of TableLens using the sort function on the "Horsepower" column [15]

2.3.4 Encode: show me a different representation

Encode techniques enable users to alter the fundamental visual representation of the data including visual appearance (e.g., color, size, and shape) of each data element. In information visualization systems, visual elements serve an important role not only because they can affect pre-attentive cognition but also because they are directly related to how users understand relationships and distributions of the data items. For instance, by encoding height information to

a map using a spectrum of color, users can better identify the height information without altering the spatial arrangement of the map.

2.3.5 Abstract/Elaborate: show me more or less detail

Abstract/Elaborate interaction techniques provide users with the ability to adjust the level of abstraction of a data representation. These types of interactions allow users to alter the representation from an overview down to details of individual data cases and often many levels in-between. A simple example is tool-tip interaction techniques that provide detailed information when a mouse cursor hovers over a data item.

2.3.6 Filter: show me something conditionally

Filter interaction techniques enable users to change the set of data items being presented based on some specific conditions. The user is not changing perspective on the data, just specifying conditions on which data are shown. As shown in Figure 2.6, The Attribute Explorer extends dynamic query capabilities by changing the colors of filtered data items rather than removing them from the display.

2.3.7 Connect: show me related items

Connect refers to interaction techniques that are used to highlight associations and relationships between data items that are already represented, and show hidden data items that are relevant to a specified item. When multiple views are used to show different representations of the same data set, it may be difficult to identify the corresponding item for a data case in other views. As shown in Figure 2.7, the brushing technique is used to highlight the representation of a selected data item in the other views being displayed.

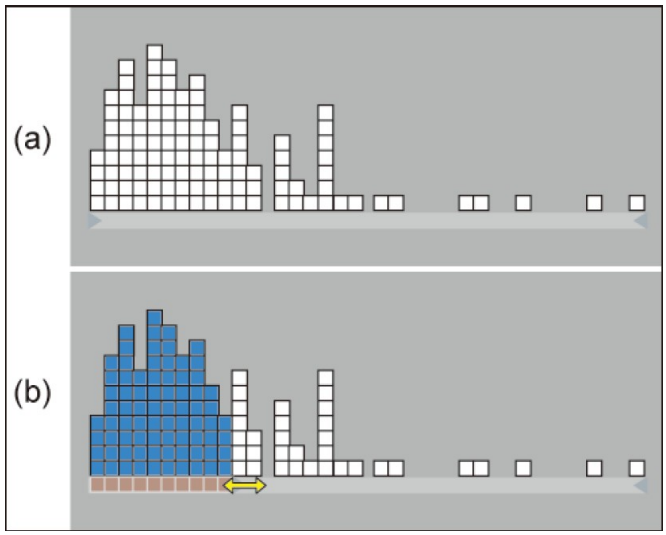


Figure 2.6: Attribute Explorer style display: (a) before changing limits and (b) after changing the lower limit [15]

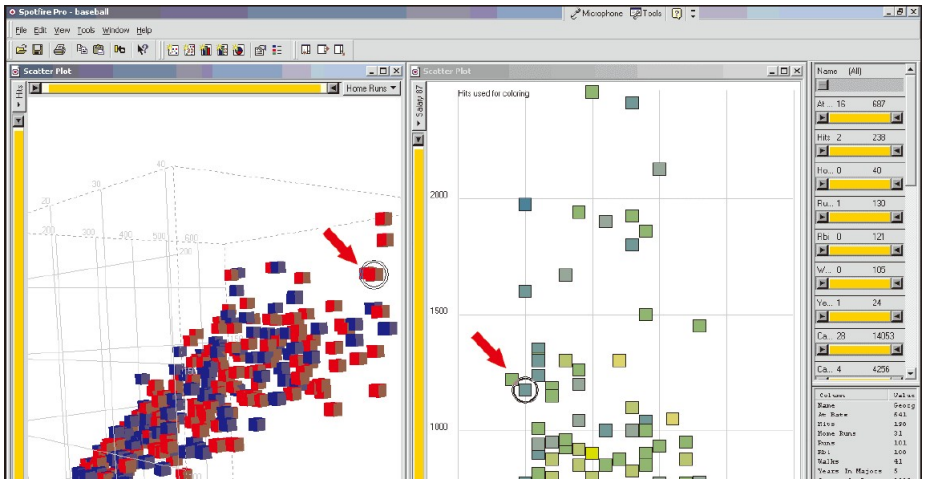


Figure 2.7: A screen shot of Spotfire showing a brushing technique [15]

Chapter 3

Data Source

Sensible DTU project keep collecting data from users by their Android smart phone. The data are gathered by a modified version of the Funf Open Sensing framework. The Funf Open Sensing Framework is an extensible sensing and data processing framework for mobile devices, which provides a variety of probes for collection, uploading, and configuration of a wide range of data signals accessible via mobile phones. For Sensible DTU project, the probes we use can be arranged in the following categories:

- Positioning:
  - Bluetooth
  - Location
  - Wi-Fi
  - Cell Tower
- Social:
  - Contact
  - SMS
  - Call log

All data is stored in JSON-structure [16], in which all of the device IDs and sensitive information are one way hashed for keeping the data anonymous. When the data is collected by the data collector on the smartphone, it is first stored in the phone's local memory. Data is archived on the phone every 300 s and

uploaded every 7200 s. The upload is only performed when the device is connected to a Wi-Fi network. As the consequent, there is a delay between the data in the phone and the data in the server.

3.1 Probes

3.1.1 Bluetooth

The Bluetooth probe scans and detects other Bluetooth devices around the user. The device scans every 300 s and records all discoverable devices within its proximity. The duration of a scan is set to 30 s. For one scan, the data contains the timestamp of the scan and a list of devices the user detects within this scan. Any Bluetooth device detected by the user will be recorded. The mac address of the device is one way hashed by the system.
Example Data:

```
{
  "TIMESTAMP": 1364768383,
  "PROBE": "edu.mit.media.funf.probe.builtin.BluetoothProbe",
  "DEVICES": [{
    "android.bluetooth.device.extra.DEVICE": {
      "mAddress": "OneWayHashed"
    },
    "android.bluetooth.device.extra.RSSI": -89,
    "android.bluetooth.device.extra.CLASS": {
      "mClass": 3801356
    },
    "android.bluetooth.device.extra.NAME": "OneWayHashed"
  }]
}
```

3.1.2 Location

The location probe records the most accurate location available for the device. The probe scans every 900 s for a period of 30 s. Furthermore, it adopts measurements made by other GPS applications, for example, if the user applies the navigator function for route directions, the probe will adopt these measurements. For one scan, the data contains the timestamp of the scan and the user's

location information which has latitude, longitude and accuracy.
Example Data:


```
{
  "TIMESTAMP": 1364767348,
  "PROBE": "edu.mit.media.funf.probe.builtin.LocationProbe",
  "LOCATION": {
    "mHasAccuracy": true,
    "mHasAltitude": false,
    "mBearing": 0.0,
    "mElapsedRealtimeNanos": 104442321000000,
    "mResults": [0.0,0.0],
    "mLatitude": 55.8110308,
    "mLat1": 0.0,
    "mDistance": 0.0,
    "mAltitude": 0.0,
    "mLat2": 0.0,
    "mLongitude": 12.5140563,
    "mExtras": {},
    "mSpeed": 0.0,
    "mInitialBearing": 0.0,
    "mAccuracy": 29.591,
    "mTime": 1364767348162,
    "mLon1": 0.0,
    "mHasSpeed": false,
    "mProvider": "network",
    "mHasBearing": false,
    "mLon2": 0.0
  }
}
```


3.1.3 Wi-Fi

The Wi-Fi probe records all of available Wi-Fi access points within the user's proximity. The probe scans every 600 s for a period of 30 s. Moreover, the probe adopts measurements made by other processes on the phone, for example, if the Wi-Fi antenna is set to always be active (even in sleep mode) then the device will constantly search for new wireless connections, and the Wi-Fi probe will adopt its measurements. For one scan, the data contains the timestamp of the scan and a list of Wi-Fi access points' information.
Example Data:

```
{
  "TIMESTAMP": 1364767281,
  "PROBE": "edu.mit.media.funf.probe.builtin.WifiProbe",
  "SCAN_RESULTS": [{
    "SSID": "Gallente Federation Network",
    "BSSID": "34:08:04:2d:57:7c",
    "level": -62,
    "timestamp": 9085540618,
    "capabilities": "[WPA-PSK-TKIP+CCMP][ESS]",
    "frequency": 2472,
    "wifiSsid": {
      "octets": {
        "count": 27,
        "buf": [71,97,108,108,101,110]
      }
    }
  }]
}
```


3.1.4 Cell Tower

The Cell Tower probe records the ID for the current cell tower the device is connected to. The probe scans every 600 s for a period of 30 s. For one scan, the data contains the timestamp of the scan and information of the cell tower. Example Data:


```
{
  "cid": 624317,
  "psc": -1,
  "PROBE": "edu.mit.media.funf.probe.builtin.CellProbe",
  "lac": 6113,
  "TIMESTAMP": 1364767325,
  "type": 1
}
```


3.1.5 Contact

The Contact probe records detailed information about the contacts available from the user's phone. The probe scans every 43200 s. Only contacts that have

changed since previous scan are recorded.
Example Data:


```
{
  "PROBE": "edu.mit.media.funf.probe.builtin.ContactProbe",
  "display_name": "{\"ONE_WAY_HASH\":\"OneWayHashed\"}",
  "custom_ringtone": "",
  "last_time_contacted": 0,
  "in_visible_group": 1,
  "times_contacted": 0,
  "contact_id": 6945,
  "send_to_voicemail": 0,
  "lookup": "389itommasobigatti",
  "TIMESTAMP": 1364770117,
  "photo_id": 63380,
  "starred": 0,
  "CONTACT_DATA": [{
    "mimetype": "vnd.android.cursor.item/name",
    "_id": 55894,
    "data11": 0,
    "is_primary": 0,
    "raw_contact_id": 6859,
    "data_version": 0,
    "data9": "",
    "data8": "",
    "is_super_primary": 0,
    "data5": "{\"ONE_WAY_HASH\":\"OneWayHashed\"}",
    "data4": "",
    "data7": "",
    "data6": "",
    "data1": "{\"ONE_WAY_HASH\":\"OneWayHashed\"}",
    "data10": 1,
    "data3": "{\"ONE_WAY_HASH\":\"OneWayHashed\"}",
    "data2": "{\"ONE_WAY_HASH\":\"OneWayHashed\"}"
  }]
}
```


3.1.6 SMS

The SMS probe records messages sent and received by the device using SMS. The probe scans every 43200 s.

Example Data:


```
{
  "body": "{\"ONE_WAY_HASH\":\"OneWayHashed\"}",
  "service_center": "+354650002305",
  "protocol": 0,
  "thread_id": 17,
  "read": false,
  "reply_path_present": false,
  "person": "",
  "status": -1,
  "address": "{\"ONE_WAY_HASH\":\"\"}",
  "date": 1364809483379,
  "locked": false,
  "type": 1,
  "subject": ""
}
```


3.1.7 Call Log

The Call log probe records in and out calls that are made by the user. Sensitive information is normalized and hashed consistently and can be compared to contacts on the device, or with other devices. The probe scans every 43200 s.
Example Data:


```
{
  "name": "{\"ONE_WAY_HASH\":\"OneWayHashed\"}",
  "numberlabel": "",
  "numbertype": "{\"ONE_WAY_HASH\":\"OneWayHashed\"}",
  "number": "{\"ONE_WAY_HASH\":\"OneWayHashed\"}",
  "date": 1364798397957,
  "duration": 3,
  "_id": 666,
  "type": 2
}
```

3.2 Fetch Data from Server

For researchers, the Sensible DTU database provides API for querying data. The API is based on http request, where the base URL is http://curie.imm.dtu.dk/sensible_outbound/v1/research/. Researchers can query the data by making a GET request to the URL with additional parameters. At first place, the query needs to specify the data from which probe. The API responses the query result from all of users in the database for the specified probe.
The probes supported by the API are as below:

- bluetooth
- location
- wifi
- contact
- sms
- call_log
- cell

All probe queries take the following parameters:

- key(mandatory): The access token.
- start(optional): A POSIX timestamp indicating the earliest time to provide data from.
- end(optional): A POSIX timestamp indicating the latest time to provide data from.
- descending(optional): A Boolean that will return the results in descending order based on time. The default order is ascending.
- limit(optional): A number for limiting the number of results.

An example for a full query:
http://curie.imm.dtu.dk/sensible_outbound/v1/research/cell?key=KEY&limit=1

All results are in JSON format in ascending order based on time. The result contains users' hashed IDs with the data from specified probe. Example data from Cell Tower probe as following:

```
{
  "OneWayHashedUserID": [{
    "cid": 624317,
    "psc": -1,
    "PROBE": "edu.mit.media.funf.probe.builtin.CellProbe",
    "lac": 6113,
    "TIMESTAMP": 1364767325,
    "type": 1
  }],
  ...
}
```

Chapter 4

Data Quality Measurement

4.1 Overall Measurement

In order to monitor the working status of the database, an overall data quality measurement is defined. For social probes, only if the user has social activity made by his or her phone the data is collected, then, there is no promise that the data from social probes should be always available. Due to the randomness of the social probes, only the quality measurements for positioning probes are taken into consideration. For positioning probes, the data collector force the Wi-Fi, Bluetooth and GPS functions keep opening. As the consequence, the positioning data should be always available while the user's phone is running.

For a given period, the data quality of a single positioning probe is measured as following:

Data available User has data from the specific probe.

No data User has no data from any positioning probe.

Missing data User has no data from the specific probe but data from another positioning probe exists.

Data error User has data error from the specific probe. Data error can be timestamp exceeding or missing key in the JSON data format.

The overall measurement is based on the single probes' quality. The purpose of the data monitoring is to check if the data collecting is working properly. Finding data error and missing data is more important than showing one probe working properly. As the consequence, data error and missing data has higher priority when measuring the overall data quality. The overall measurement is defined as following:

Data available User has data from all of probes.

No data User has no data from all of probes.

Missing data One of probes has missing data and none of the probes has data error.

Data error One of probes has data error.

4.2 Measurement for single probe

4.2.1 Bluetooth

The purpose of collecting Bluetooth data is to check if a user meets other users in proximity. For a given time period, the Bluetooth data is measured in three ways as following:

User is scanning User has scan data from the Bluetooth probe. It is possible that no device is detected in the scan.

User is scanned by other user The user's device ID appears in the device list of another user's scan data.

User is observed The user's device is once scanned by another user before the given time period.

The quality measurement also takes data from other probes into consideration. Within a given time period, if the user has no data from Bluetooth probe but data from other probes, it means the data from Bluetooth probe is missing. As the result, the Bluetooth data quality is divided into six categories as following:

1. User scanning and being scanned by other user
2. User scanning and observed but not being scanned
3. User scanning but not observed
4. User being scanned but has no data from Bluetooth probe
5. User has no data and not being scanned
6. User is not scanning and not being scanned but data from other probes exists

4.2.2 Location

For research purpose, the more GPS data points are recorded by the user, the more accurate movement path can be generated for the user. The data quality of location probe is measured by the number of data point within each scan. Within the 30 s scan period, there should be several data point recorded from the location probe. The location data quality is divided into four categories as following:

1. Good data quality: User has more than 5 data points from location probe.
2. Normal data quality: User has at least 1 data point from location probe.
3. No data: User has no data from location probe.
4. Data missing: User has no data from location probe but data from other probe exists.

When measure the data quality for more than one scan, the period more than 1800 s, the data quality is normalized from no data to more than 5 data points. Within a given period, if each scan has more than 5 data points from location probe, the data quality is considered good and belongs to category 1. If each scan has at least one data point but not all scans have more than 5 data points, the data quality is between category 1 and category 2 based on the number of scans with more than 5 data points over the number of all scans. For example, the normalize data quality from category 2 to category 1 is [0, 1], if there are two scans belongs to category 2 within the given time period and only one of them belongs to category 1, the data quality for this period is 0.5. The data quality between category 2 and category 3 is measured by the same way.

4.2.3 Wi-Fi

For research, the more Wi-Fi access points are recorded by the user, the more information is available. The data quality of Wi-Fi probe is measured by the number of Wi-Fi access point within each scan. The Wi-Fi data quality is divided into five categories as following:

1. Good data quality: User has more than 10 Wi-Fi access points from Wi-Fi probe.
2. Normal data quality: User has at least 1 Wi-Fi access point from Wi-Fi probe.
3. Bad data quality: User has no Wi-Fi access point from Wi-Fi probe.
4. No data: User has no data from Wi-Fi probe.
5. Data missing: User has no data from Wi-Fi probe but data from other probe exists.

When measure the data quality for more than one scan, the period more than 1200 s, the data quality is normalized the same way as the location probe.

4.2.4 Cell Tower

When the user's phone is running with online mode, the data from Cell Tower probe should be always available. The quality measurement is based on whether the user has data from cell tower probe. Compare the data to other probe and check if the probe has missing data. Within a given time period, if the user has no data from Cell Tower probe but data from other probes, it means the data from Cell Tower probe is missing. The Cell Tower data quality is divided into three categories as following:

1. Good data quality: User has data from Cell Tower probe.
2. No data: User has no data from Cell Tower probe.
3. Data missing: User has no data from Cell Tower probe but data from other probe exists.

Chapter 5

Design

5.1 System

Due to the objective of the data reporting system is providing an easy access to the database's working status, developing the application with Client/Server structure is not a wise idea. On the other hand, a web application based on browser is a good solution, which makes the system available for most devices with browser.

Consider a simple use case of the system: user first open the web page of the system, select one aspect of data quality he or she interested in, then choose a period of time and get the visualization of users' data quality. From the use case, there are several basic components for the system.

- User interface

  Navigation Let the user switch between different aspects of data quality easily.

  Date picker The user should give a period of time to the system for visualization. It is impracticable to visualize the data quality for whole database with good detail. Along with the size of data for

visualization grows the level of detail decreases. A date picker is a simple approach for selecting a period of time. In addition, it is easy to put limits on the date picker.

Visualization A component that is responsible for visualizing the data quality chosen by the user.

- Application server

Request handler Handle the request from the user and render the web page of the system.

Data handler Fetch the data requested by the user from the database and generate the data quality.

The components discussed above only meet the requirements for a general use case. There are some components for improving the user's experience.

- User interface

Ordering selector An ordering function is useful in data visualization which makes the user easier for finding the pattern or correlation of the data.

ID list After the user get the visualization of the data quality, it is normal to find some participants' IDs for further usage, such as sending notifications to the participants who have a lot of low quality data. The ID list gives the user a list of selected participants' IDs in text format when the user interacts with the visualization.

A mockup of the general user interface with UI components discussed above is show in Figure 5.1.

5.2 Visualization

Data quality visualization is the main component of the system. Consider the data structure of the Sensible DTU database, there is a lot of participants in the database and each of them has a big amount of data. The more detail we want to get from the visualization, the more space is needed for the visualization. While the space of the browser is limited, it is impossible to make visualization with unlimited size. Taking the capability of the browser and user experience into account, it is important to put as much information as possible into the

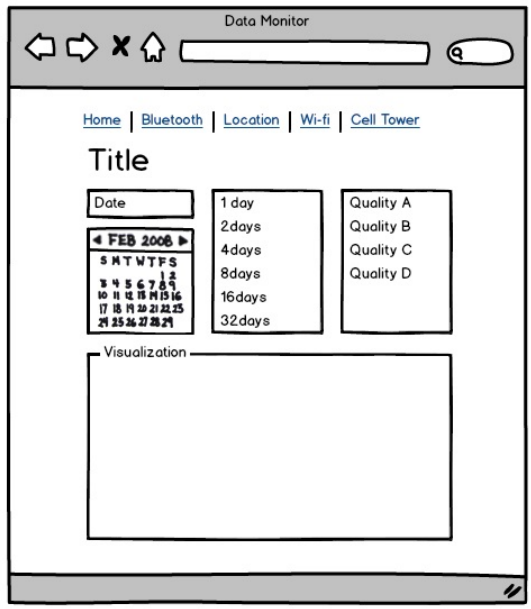


Figure 5.1: Mockup of the general user interface

visualization with limited size. In order to make full use of the space, a matrix is a good solution, which can fill almost all available space with information. In the matrix, we need to identify different participants' information and the time information, so the location dimension can be used for presenting such information. Furthermore, another dimension is needed for representing the data quality information, color is a fantastic representation method for enormous sets of data. We can identify many gradations and shades of color and can see differences in a high resolution.

Figure 5.2 shows an example of the matrix visualization, each row represents one participant's data quality and each column is one hour time bin. The data quality is encoded as colors, which makes it easy to distinguish different data qualities from the visualization. In the matrix, one participant's data quality only takes a small space, so we can put as many participants as possible into the visualization before we cannot distinguish two different participants'. For example, if the height of a row is 3 pixels, the height of the visualization for 135 participants is less than 400 pixels, which is possible to be displayed in one screen. Furthermore, it is possible to decrease the width of the column and put more columns into the matrix, which enable the visualization to contain data for longer period but with the same level of detail.

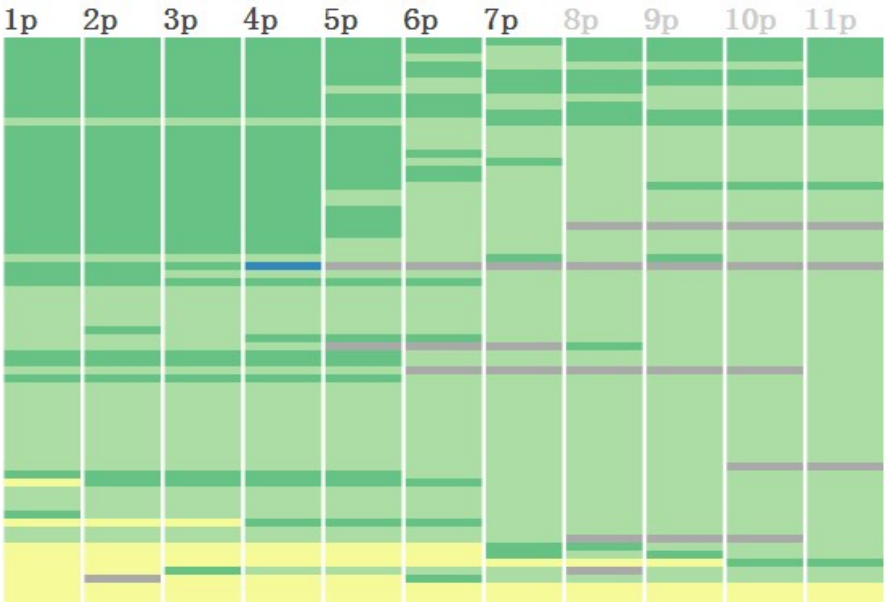


Figure 5.2: An example of data quality matrix. Each row represents one participant, each column represents one hour time bin and colors represent the data qualities.

The matrix is useful for visualize the data quality for a single probe. While there is also an overview of the data quality from all positioning probes needed to be visualized, the matrix can only convey the quality information by color which is not enough for the overview. It is possible to present the overall quality of the positioning probes by the matrix, from the definition of the overall quality, it is easy to find which participant has missing data or data error but it is impossible to know how much data is missing. Different probes have different scanning rate, so the amount of data is different from different probe within the same time bin. For example, if a participant's data collector is working properly within half hour, there should be 6 scans from Bluetooth probe, 2 scans from Location probe, 4 scans from Wi-Fi probe and 4 scans from Cell Tower probe. Missing one scan from Bluetooth probe should be considered differently with missing one scan from Location probe. From another aspect, if a participant's data quality for one time bin is marked as missing data, it is possible to miss data from one probe or three probes while it is presented the same in the matrix. As the consequence, there should be an additional dimension for visualizing the amount of data within one time bin.

Follow the idea of the matrix, the amount of data within one time bin can be encoded with other visual variable. Since the position and color is already used by the matrix, size is a good choice for the new dimension, which is easy to be visualized and distinguished. In the matrix, the size of each data quality is minimized so that it is necessary to increase the size of a single data quality for making it possible for the user to distinguish the difference between sizes. Change the shape of the single data quality from rectangle to circle which is easier to distinguish with different sizes. Figure 5.3 shows an example of visualizing the overview of data quality by bubbles. The amount of data is encoded into the size of bubble, smaller bubble means less data. Combine the color and size of the bubble, we can find much more information than matrix. For instance, a small green bubble means the participant has no missing data within the time bin but only a small amount of data is uploaded to the server. On the other hand, a big red bubble means the participants' data collector is working properly for the most of time within the time bin while there is a small amount of missing data.

By using matrix and bubbles, more information is available than using only matrix, but there is still not much information within a time bin. Sometimes, it is important to look into the detail of a time bin because the user may visualize the data quality for a whole month and the time bin can be increased to half day. It is possible to let users decrease the length of period of time and find the detail of the time bin by their selves, which leads to a bad user experience. Furthermore, if the time bin is already in the minimum size, it still larger than the scan rate of the probe. As the result, a detail view of one participants' data quality within one time bin is added. User can interact with the visualization, clicking

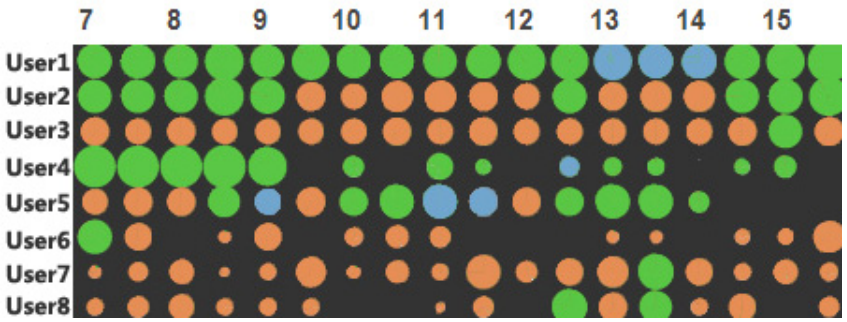


Figure 5.3: Visualizing the overview of data quality by bubbles. Each row represents one participant, each column represents half hour time bin, colors represent the data qualities and the size of bubbles represents the amount of data within the time bin.

a bubble and then get the detail view. The detail view contains a pie chart for the statistic information of the time bin and a timeline for all probes. From the timeline, user can easily compare the data quality from different probes. In Figure 5.4, there is a mockup for the detail view.

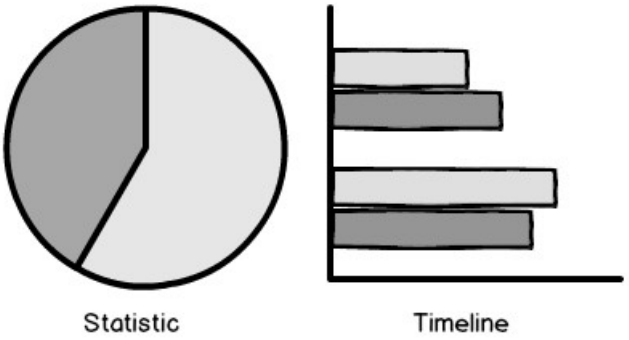


Figure 5.4: Mockup for detail information visualization of a single time bin

5.3 Animation

Animation is an important part of the visualization interaction. During the animation, viewers have more time to retain their impression of the old view. If the change from one view to another is presented by smooth transitions instead

of discrete jumps, viewers can get better understanding of relation between the old and new views. A smooth and proper animation can make the visualization vivid while a repeat of static visualization in the same format makes viewers bored. In the system, there are a lot of change between different views, such as generating a new matrix and reordering the matrix. Using proper animation can improve the user experience significantly. Several styles of animation are discussed as following.

5.3.1 Flip Animation

Flip animation makes the object working like a coin with two sides. The transition of the animation is flipping the old side of the object to the new side. Figure 5.5 shows a flip animation applied on the matrix. Flipping each rectangle by different time makes the transition vivid and gives viewers more time during the animation.

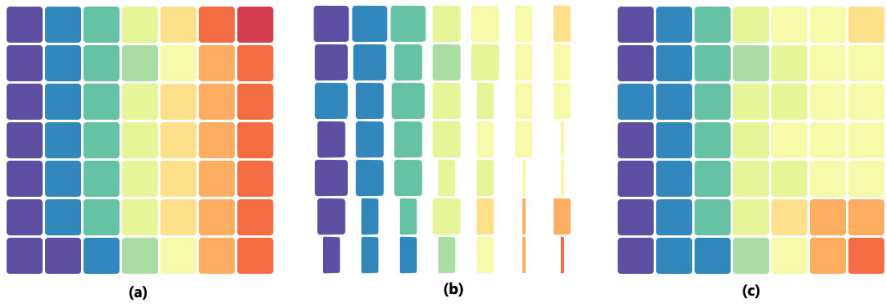


Figure 5.5: Flip animation from Trulia Trends [17]

In the system, there is a compact and big matrix which contains more than thousands rectangles. Flipping the rectangle one by one requires a strong capability of the browser. If the capability of the browser is not strong enough, the animation will not be smooth which decreases the user experience. As a trade-off, make each row as a coin then we can flip the whole matrix. Flipping each row one by one decreases the requirement of browser's capability, which leads to a proper animation for the big matrix. Figure 5.6 shows the example of the flip animation.

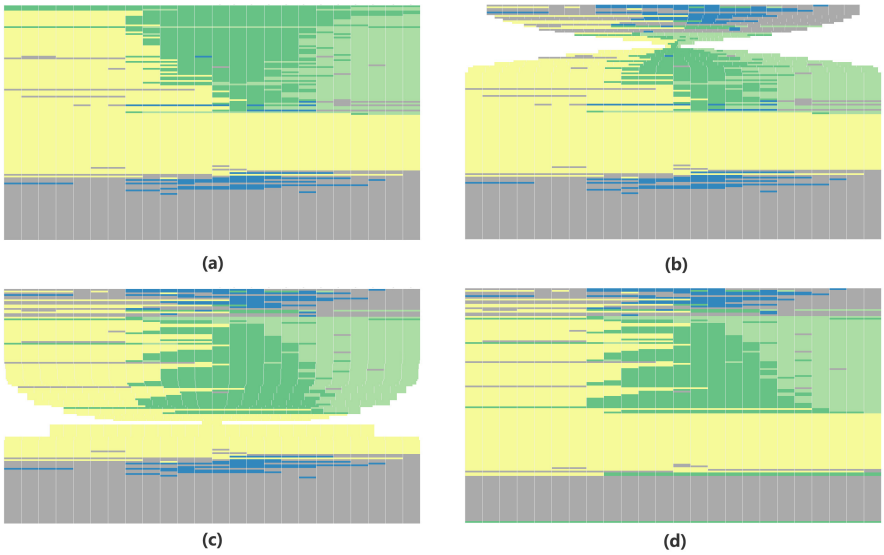


Figure 5.6: Screen shots of flip animation on matrix visualization

5.3.2 Zipper Animation

Zipper animation is useful for generate the matrix. The animation works as a zipper, all of the rectangles in the matrix start on the central top position and then each row move low to reach the right position. Along with the movements of the rows, the rectangles move to both sides for the right position. Figure 5.7 shows the example of the zipper animation. The advantage of zipper animation is that it can be applied on an empty space while flip animation needs an old side.

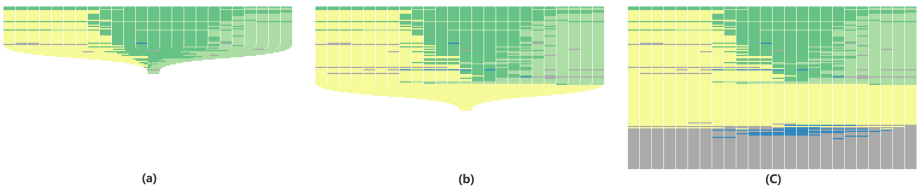


Figure 5.7: Screen shots of zipper animation on matrix visualization

5.3.3 Reorder Animation

Reorder animation is useful for reordering the matrix. It moves all of rows from old positions to new positions directly by a smooth transition, which gives viewers a link between the old order and the new order. Figure 5.8 shows the example of the reorder animation.

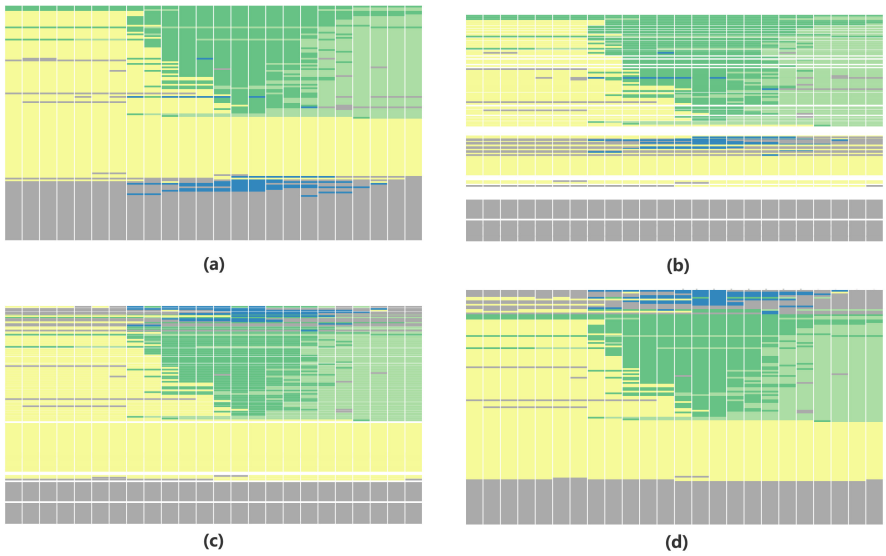


Figure 5.8: Screen shots of reorder animation on matrix visualization

5.4 Data Collecting

For a data reporting system, the data is the most important thing. As mentioned before, there is an API for query the Sensible DTU database while it only supports querying all users' data from one specific probe. Using the API directly as the data resource is not a wise idea. For example, consider the situation when we want to check all users' data quality for one day. Currently, there are 135 users in the database. The size of query result for Wi-Fi probe is about 500 MB, which is a big amount of data. The reason is that for each user there may be several Wi-Fi access points in each scan and the query result contains detail information for each Wi-Fi access point. It is not a short time for waiting the application server downloading the data, which will lead to a bad user experience. Even if we assume there is a high speed connection between

the application server and the API, consider another situation, when we want to check all users' data quality for a month, the application server needs to download 15 GB data from the API which is unreasonable to do for each data report.

After a trade-off between time efficiency and space efficiency, the solution is setting up a separate database for the data reporting system which contains only quality information from users. Figure 5.9 describes the database of the data reporting system. For each user entity, the name is the hashed id in the Sensible DTU database and the observed attribute is the timestamp of first time the user appeared in other user's Bluetooth scans. Furthermore, every user entity has zero or more entities which are data quality information for different probes. Each entity other than user has a time attribute that is the timestamp of the data quality information, which is based on the scan rates of different probes. Bluetooth entity has scanning and scanned attributes that are Boolean values as quality measurements. Location and Wifi entities records the number of data points and access points in count attribute. The probe attribute in Error entity specifies the source of the error.

Due to the database of the data reporting system is separated from the Sensible DTU database, an update mechanism is necessary. Since the users only update their data to Sensible DTU database when they have Wi-Fi connection, the database for data quality is not necessarily up to date in real time. In common, the data in Sensible DTU database has one day's delay from the data recorded in user's phone. As the consequence, the data reporting system database can be updated periodically and keep the data quality information up to date. The update mechanism is designed as following:

- For each hour, the data reporting system gets one hour data of 24 hours ago from the Sensible DTU database for update.
- For each day, the data reporting system gets data of the last day from the Sensible DTU database for update.
- For each week, the data reporting system gets data of last month from the Sensible DTU database for update.

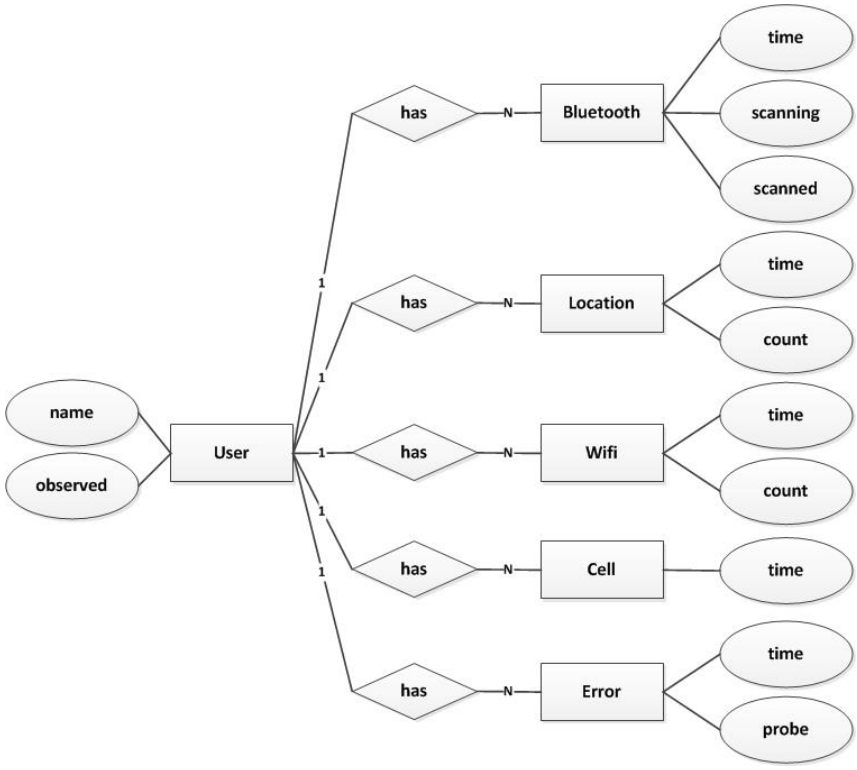


Figure 5.9: Entity–relationship diagram for the data reporting system database

Chapter 6

Implementation

The design from previous chapter is the guideline for developing the system. This chapter first describes a prototype of the system which follows the original design. During the develop process, there are several improvements applied on the system which makes the system different from the original design. The result of the system has better usability and user experience than the prototype.

As described in the previous chapter, the web application is based on browser and the visualization is generated in the browser, so the support of visualization libraries in browser is needed. With the advance of browser and computational power, many SVG based JavaScript visualization libraries have appeared, such as data-driven documents (D3.js) [18] and Processing.js [19]. D3.js library is chosen for the system. The idea of the D3.js is to provide an easy approach for efficient manipulation of documents based on data. This makes the development of visualization in browser much easier. The web application also uses jQuery [20] and jQuery UI [21] library which can easily generate UI components and handle interactions, effects, animation and AJAX.

6.1 Prototype

Before implementing the system, a prototype is made for testing the original design and finding potential improvements. The prototype is implemented based on Google app engine [22] which allows users easily deploy their web applications on Google's infrastructure. Web application based on Google app engine is easy to develop. The deployment and maintenance is supported by Google's Cloud Services.

The prototype only implements the data reporting system for the Bluetooth probe which contains basic user interactions and visualizations. Figure 6.1 is a screen shot of the general user interface for the prototype. Due to the prototype is only for one probe, there is no navigation implemented. The prototype keeps downloading Bluetooth data from Sensible DTU database periodically and updating the data quality database which ensures the prototype's database is up to date.

The period of time selection is implemented by a date picker and a radio button group. By the date picker, the user can choose the last day of the visualization. The date can be changed day by day with the arrow buttons, and user can click the calendar icon to get a popup calendar (Figure 6.2) for changing the date directly. The radio button group supports the switch between different lengths of period from 1 day to 32 days. When the date or the length of period is changed, the application will query new data from the server and generate a new matrix.

When the page is opened at first time, the default date is yesterday and the length of period is 1day. After the page is loaded to the browser, it starts to request data for the visualization by AJAX. Before the visualization is generated, the visualization area shows a loading animation (Figure 6.3) which indicates the application is loading the data. During the loading, all buttons are disable until the visualization is generated which prevents the application from sending multiple requests to the server.

When the data is loaded, the matrix will be generated by a zipper animation. On top of the matrix, there is a header showing the time of the columns in which the hours of daytime and night are marked as different colors. The number of rows is the number of participants in the database. The number of columns is based on the length of period. The minimum time bin for one column is one hour. Consider the minimum width of the column, if the chosen length of period is less than 8 days, the size of time bins is not changed so that the number of columns will be increased. When the chosen length of period is equivalent or more than 8 days, the width of column is already the minimum value so that the

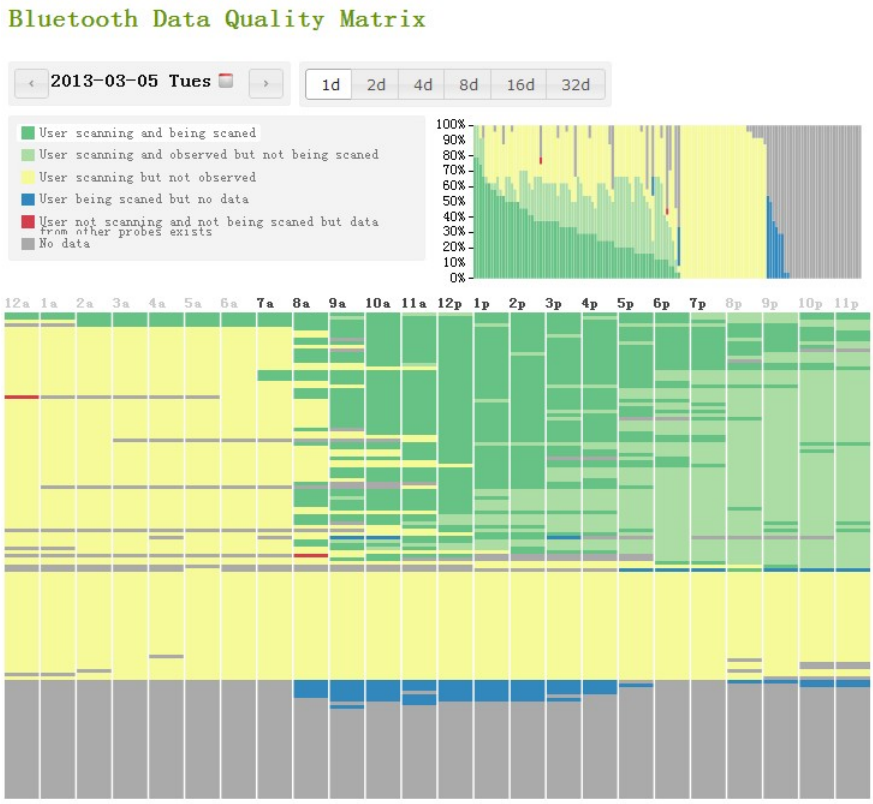


Figure 6.1: General user interface of the prototype

Figure 6.2: Screen shot of the popup calendar

Figure 6.3: Screen shot of the loading animation

number of columns is not increased anymore and the time bin becomes larger. Figure 6.4 shows the relation between the length of period and the number of columns.

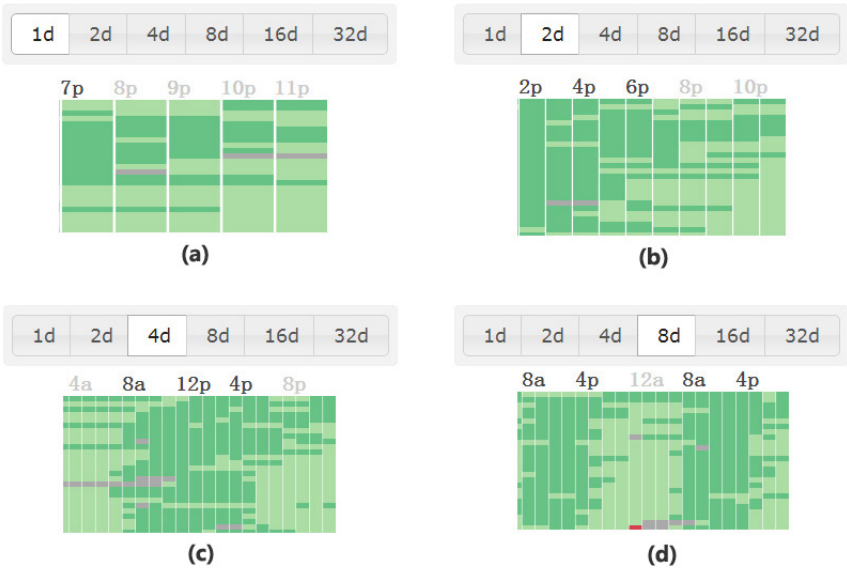


Figure 6.4: Length of period changes the number of columns. (a) Matrix visualization of 1 day (b) Matrix visualization of 2 days (c) Matrix visualization of 4 days (d) Matrix visualization of 8 days

For the ordering operation, user can click the description of data qualities to change the order of the matrix with a reorder animation. Next to the data quality description, the bar chart shows the statistic information of the matrix, each bar represents one participant. The order of the bar chart is the same as the matrix. When the order of the matrix is changed, the order of bar chart is also changed by reorder animation. User is available to interact with the matrix by selecting an area of the matrix. After the selection, the bar chart will change to the view of statistic information for the selected area as shown in Figure 6.5. The user can also get a list of selected participants' IDs.

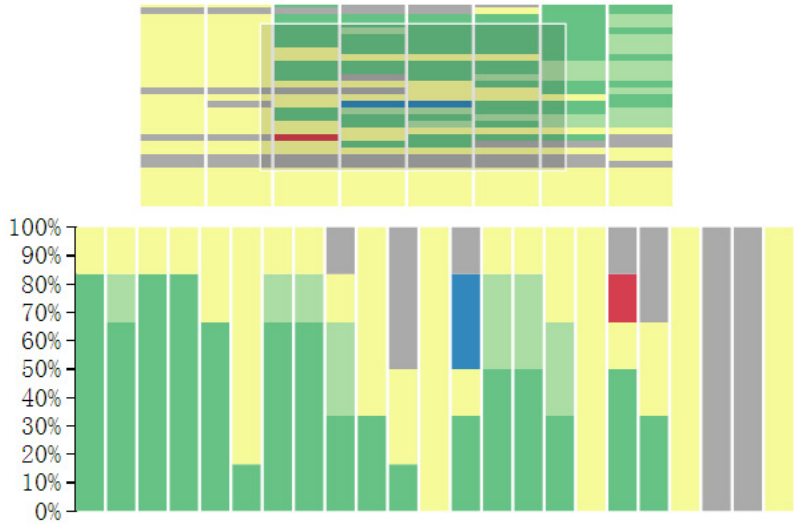


Figure 6.5: Screen shot of the selecting interaction. The first box is the selection on the matrix visualization. The second box is a bar chart visualizing the statistic information of selected area, each column is one participants.

From the feedback of the prototype, it meets the basic goal of the data reporting system since there are a lot of things to improve. The most important limitation is the size of the matrix. In the prototype, it always generates the data qualities of all participants in the database. For 135 participants, the loading time and browser capability is affordable. While the number of participants is going to increase to more than 1000, generating the matrix for all participants at once is no longer a wise idea. If the user wants to check the data qualities for one month, the server needs to query a big amount of data for the visualization which leads to a bad user experience for the long waiting time. Furthermore, there will be a matrix with more than 100,000 rectangles, considering the capability of the browser, the animation will not work as smooth as we expect. As the consequence, a solution for limiting the number of participants in the matrix is necessary.

6.2 System Architecture

Based on the prototype, the final implementation is called Sensible DTU Data Monitor which is a web application based on browser as the design. The back-

end of the system is implemented in Django [23] framework. Django is a free and open Python web framework which follows the model–view–controller architectural pattern. The framework's goal is to ease the creation of complex, database-driven web apps with less code.

Figure 6.6 shows the system architecture of Sensible DTU Data Monitor. The user client is based on browser and communication between user and server is through the Internet. The system consists of an application server and a database server. The application server handles the user's request, which contains generating the web page and data queries. In addition, the application server also handles the data updates between the Sensible DTU database and the data quality database for the system. The application server contains 6 components, the data collector is responsible for the periodically updates from the Sensible DTU database. The communication between the data collector and Sensible DTU database API is through the Internet. The overview component generates the home page of the system and handles the data for overall data quality visualization of positioning probes. The rest of the components handle the web pages for each single probe and the data for the visualization.

6.3 Frontend

Base on the prototype, the final implementation has several improvements. Figure 6.7 shows the home page of the Sensible DTU Data Monitor which visualizes the overall data quality of the positioning probes.

Different from the prototype, the date picker and the period selection panel is combined together. For the period selection, the length of period is changed to 1 day, 2 days, 4 days, 1 week, 2 weeks and 1 month, which aligns to the common sense. Due to the home page is to visualize the overall data quality, a filter panel is added which enables the user to switch the view between the overall data quality and single probes. Furthermore, an accordion replaces the matrix visualization. The accordion consists of 3 tabs that are overview, user list and detail. The overview tab contains statistic information of all participants for the chosen period of time. The user list tab shows a user matrix that contains all of participants in the database, and then the user is able to select up to 50 participants to generate the matrix visualization of data quality. The purpose of the new structure is to overcome the limitation of the matrix size. The size of the matrix visualization is based on the user's selection and has a limit of 50 participants, which ensures a reasonable loading time of the matrix visualization and the smooth animation in the browser. For the accordion, this is an UI component that can save the space of web page. Within each tab, there is a

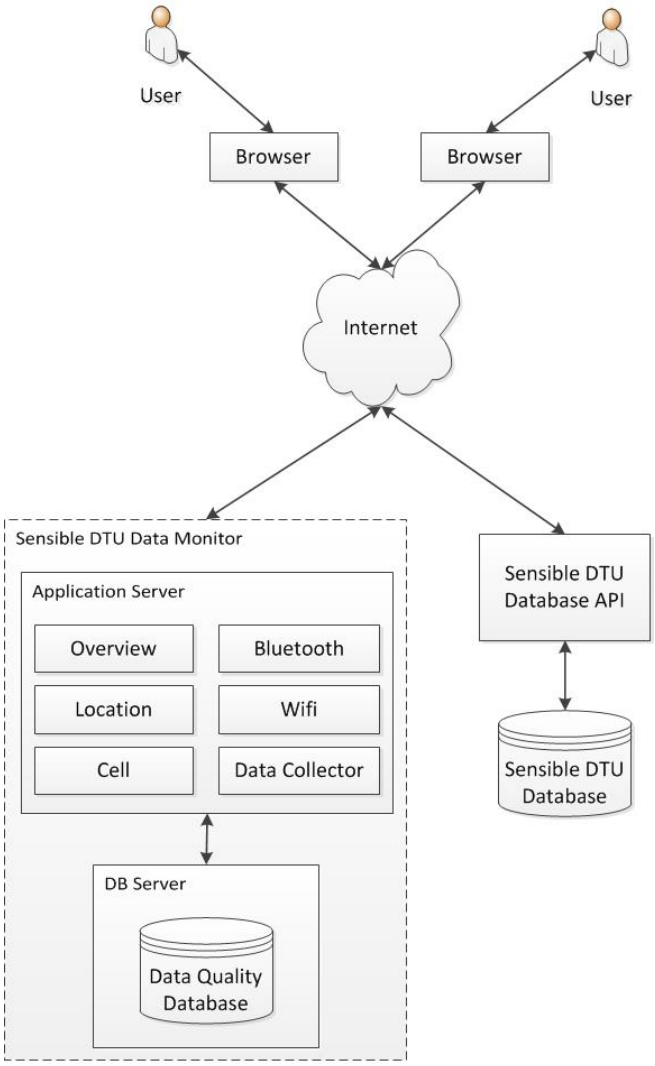


Figure 6.6: System architecture of Sensible DTU Data Monitor. Users get access to the system through internet. The system contains an application server for handling the user requests and updating the data quality database by using the Sensible DTU Database API. The communication between the system and the API is through internet.

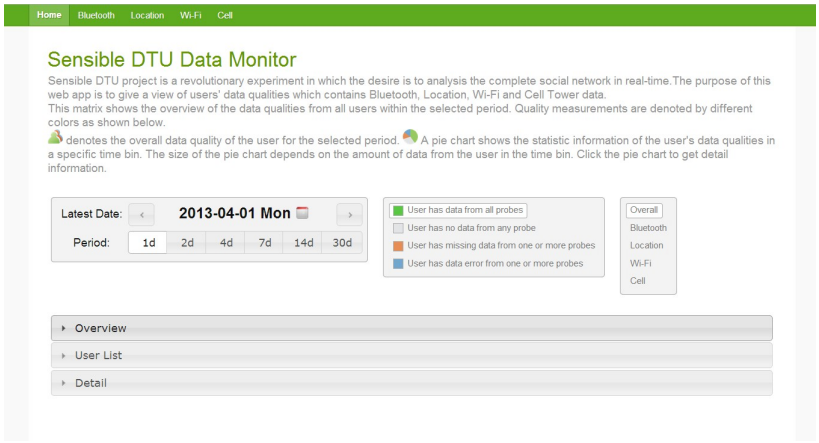


Figure 6.7: Screen shot of the home page

group of visualizations for different purpose, showing all of the visualizations at the same time will make the web page too long. The accordion makes only one group of visualizations is display at the same time and enables the user to toggle which group of visualizations to display.

When the web is loaded at the first time, the visualizations in overview and user list tab are available based on the time of date picker. Since the detail tab is empty at beginning, the matrix visualization is generated by the user's interaction with the user list. In Figure 6.8, there is a flow diagram for generating the web page of the system. Instead of generating the web page and visualizations in the first two tabs at the same time, the system only generates the web page for the user's request. After the web page is loaded in the browser, the client sends another request for generating the data for visualizations by AJAX. Before the visualization is generated, there is loading animations working as the place holder of the visualization. The reason behind this flow is that generating the data for overview and user list visualizations may be a long time. If the period of the visualization is one month, processing all of the data for more than 1000 participants takes a long time. As the consequence, if loading the web page and visualization at the same time, there may be a long waiting time before anything appearing in the browser, which may makes the user confused with whether the system is working properly. Due to the most part of web page is static, the user can get the web page in a short time after the request. Then, the loading animation indicates that the system is working and some time is needed for generating the visualization. This leads to a good user experience with the system even if the loading time is a bit longer than the user expected.

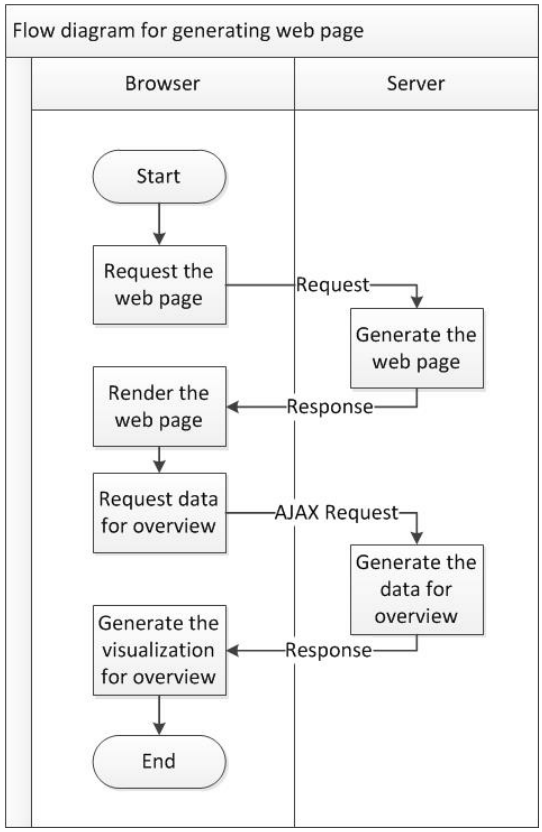


Figure 6.8: Flow diagram for generating web page

6.3.1 Overview

Figure 6.9 show the screen shot of the overview tab, which contains a line chart and a pie chart. Following the color definition next to the date picker, the colors in the overview represents groups of participants with different data qualities.

Green User has data from all probes

Grey User has no data from any probe

Orange User has missing data from one or more probes

Blue User has data error from one or more probes

The overview gives statistic information about participants' data quality. In the line chart, the x-axis represents the time and the y-axis represents the number of participants. For each time bin, each participant is classified into one of the data qualities, so each line shows the changes of the number of participants with the specific data quality over time. For the whole period of time, each participant also get a total data quality based on the data qualities for every time bins. Then, the pie chart shows the percentage information about the participants' total data qualities. From the overview tab, the user can easily get the overview of the database working status.

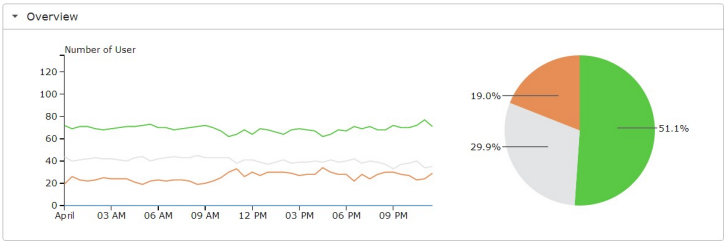


Figure 6.9: Screen shot of the overview statistic information. The line chart visualizes the number of participants with different data qualities. The pie chart visualizes the percentage information about the participants' total data qualities.

6.3.2 User List

Figure 6.10 shows the screen shots of the user list, which contains a user matrix on the left and a user list on the right. The user list is an important feature in

the final implementation of the system, which builds a bridge from the overview to the detail information.

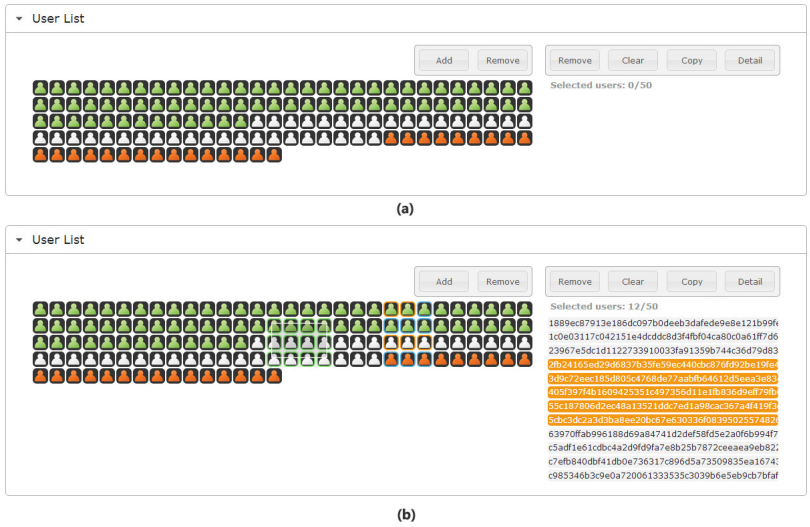


Figure 6.10: Screen shots of the user list. (a) The left part is the user matrix that consists of user icons for all participants. The color of the user icon is the total data quality of the user. (b) The right part is the user list containing the participants added from the user matrix.

The user matrix contains all of participants in the database and each participant is represented as a user icon. The color of the user icon is based on the participant's total data quality of the chosen period of time. There are 30 user icons in each row of the user matrix. Even if the number of participants becomes more than 1000, the user matrix can be visualized in a reasonable space. From the user matrix, the user is able to select a group of participants as shown in the second screen of Figure 6.10. The icons of participants selected by user are highlighted in green. Then, using the menu above the user matrix, the user can add the selected participants into the user list. Once the participant is added to the user list, the color of the icon is highlighted in blue.

In the user list, the participant is represented as a list item that contains the hashed ID of the participant. The list items in the user list are selectable as shown in the second screen shot of Figure 6.10, and the user icons of selected participants are highlighted in orange. In the menu above the user list, there are several operations available. The user can remove the selected list item from the user list. On the other side, the user can also remove participants

by another way. After selecting a group of icons in the user matrix, using the "remov" button in the user matrix menu, all of the selected participants in the user matrix will be removed from the user list. The copy operation enables the user to copy the participants' IDs from the user list as text. While accessing the system clipboard directly from browser is not safe, the "copy" operation is implemented as a prompt window with the text of participants' IDs, in which the user can copy the text manually. For the "detail" operation, the system will generate the matrix visualization for the participants in the user list.

6.3.3 Visualization

By using the "detail" operation in the user list, the matrix visualization for a group of participants is generated in the detail tab. Figure 6.11 shows the matrix visualization of the overall data quality. In the matrix, instead of using bubbles in the original design, the data quality information is represented by small pie chart. When using bubbles, the color of the bubble is based on the overall quality of the time bin. As the result, even if there is only a small fraction of data missing, the whole bubble is marked as missing data. Then, there will be a lot of big bubbles marked as missing data, which may make the user get a feeling of most data in the database is not good. By using the pie chart, the amount of missing data is clearly visualized. The size of the pie chart is the same as the bubble which is based on the amount of data within the time bin. If there is a lot of missing data or no data within the time bin, the size of the bar chart will be small. Then the big pie chart always has big fraction of good data quality, which leads to a good view for the overall viewpoint. Combine the size and color, user can easily find which pie chart representing a bad data quality. One additional feature is that a small user icon is added in front of each row representing the data quality for the whole period of time.

Clicking on one of the pie chart, the user can get a detial view of the data quality within the time bin. The detail view is generated in a popup dialog so that no more space is needed for the new visualizations and the old visualiztion is not affected. As shown in Figure 6.12 and Figure 6.13, the detail view consist of a title indicating the time bin, the participant's ID and a tab panel for visualizations. The tab panel contains two tabs, one is the visualization of statistic information, the other tab is the timeline of each single probes. In the statistic tab, the statistic information of each probe is visualized by a donut chart which shows the fraction of different data qualities. Inside the donut chart, one icon represents the probe that enables the user to identify the probe easily. In the timeline tab, each probe has a timeline for the time bin. The number of data points in each timeline is based on the scan rate of the probe. For example, the scan rate of Bluetooth probe is 300 seconds, so there is 6

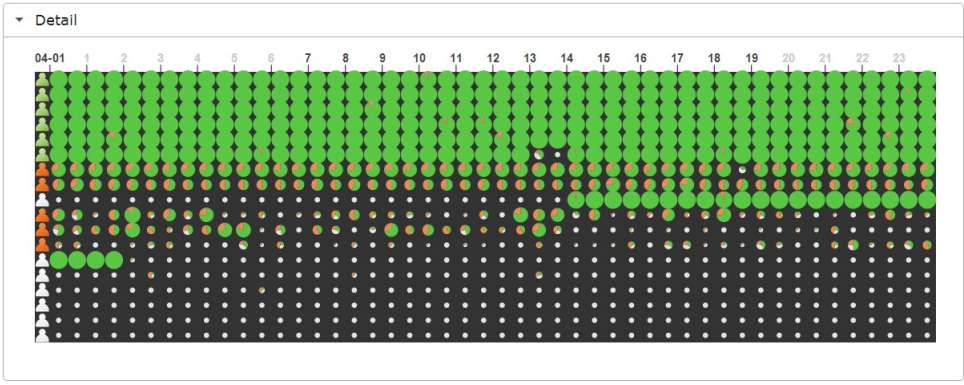


Figure 6.11: Screen shot of the visualization for overall data quality. Each row represents a participant in which the user icon shows the total data quality. Each column represents half hour time bin. The pie chart visualizes the data quality distribution within the time bin.

data point in the timeline of half hour. From the timeline, the user can get the data quality information in highest resolution. Comparing the timelines from different probes, the user is able to find the reason why the data of a specific time is considered as missing data.

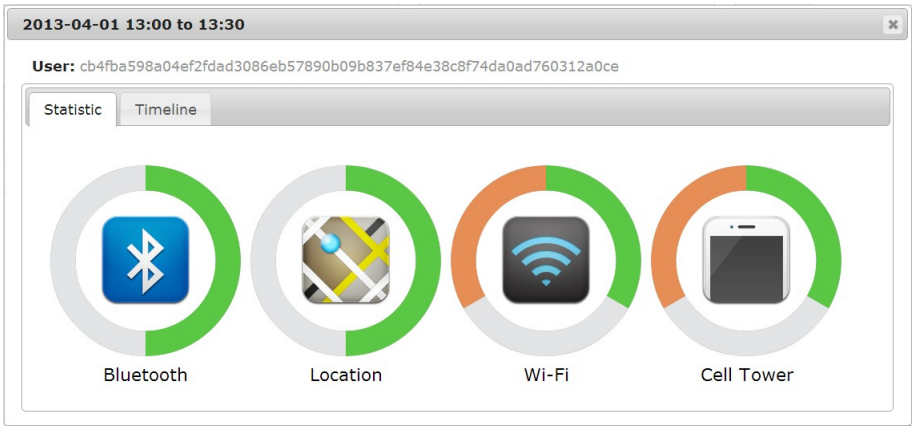


Figure 6.12: Screen shot of the statistic tab in detail view. The donut charts represent the data quality distribution of the time bin for all probes.

Figure 6.14 shows the flow diagrams of generating the matrix visualization and

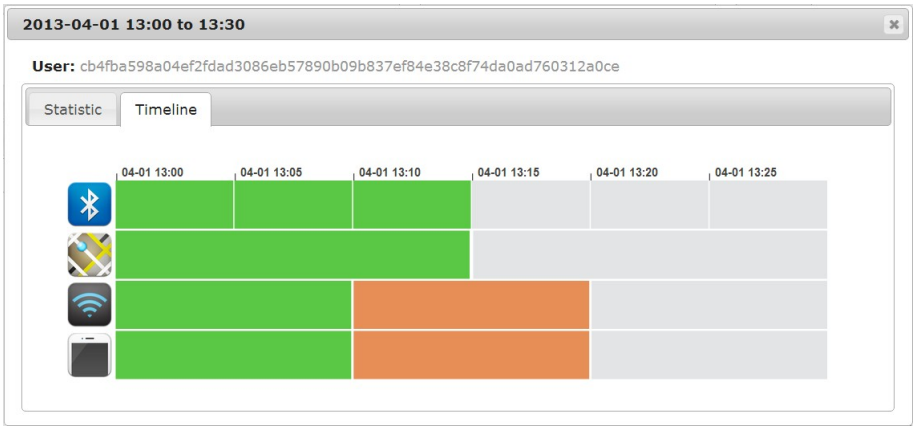


Figure 6.13: Screen shot of the timeline tab in detail view. The timelines visualize the data quality information in the highest resolution for all probes.

the detail view. The data for generating the visualization is loaded from the server by AJAX. While loading the data for generating the matrix visualization, the interactions with the UI components are disabled until the data is loaded to the browser. The purpose is to prevent the user sending multiple requests to the server that may make the system go into a wrong status. For example, if the user changes the period of time before a visualization being generated, a new request of data query is sent to server by AJAX. By AJAX, there is no promise that the first response always arrive before the second one. If the second response arrives before the first one, the system generates the visualization for the new period of time. Then, the first response arrives and the system generates the old visualization. As the result, the user gets the old visualization but the right visualization of the user's choice is missing. Generating the detail view follows the same idea. When the dialog is created, the user can only interact with the UI component inside the dialog.

For the matrix visualization of single probe's data quality, the final implementation of the system uses the same matrix from the prototype. The data quality measurements for Bluetooth probe and Cell Tower probe are qualitative while the measurements for Location probe and Wi-Fi probe are quantitative. So, the way of color encoding for Location probe and Wi-Fi probe is different from the prototype.

In Figure 6.15, there are 3 examples of the matrix visualization for Location probe. From the data quality definition of the Location probe, when the participant has data from Location probe, the data quality is not represented by

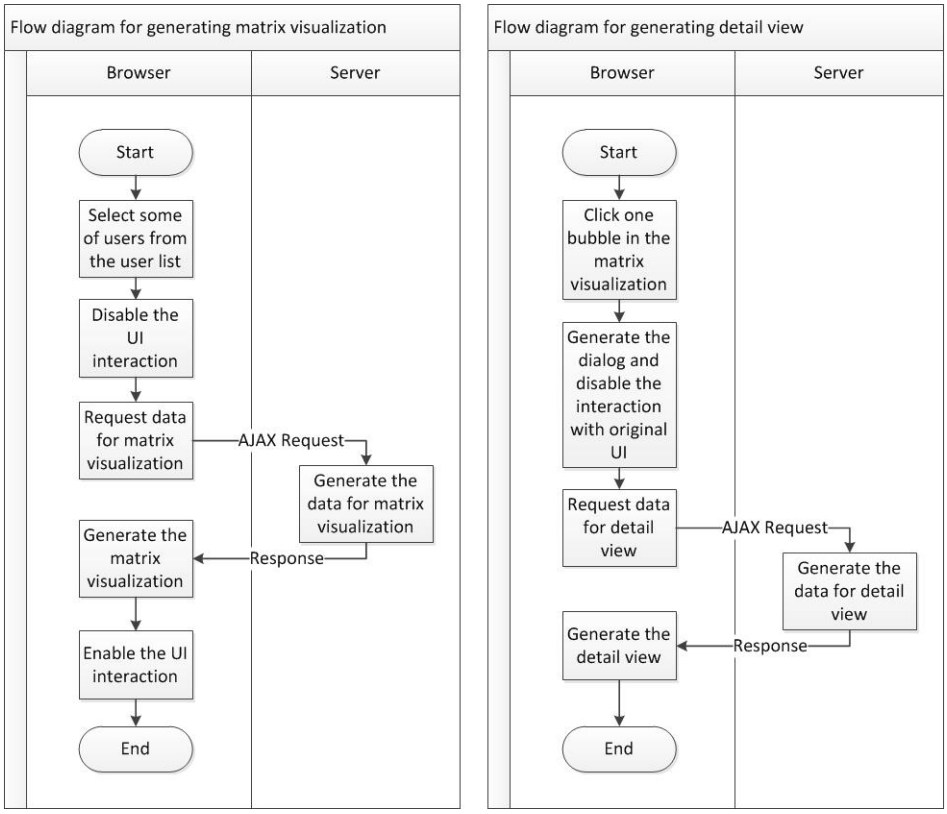


Figure 6.14: Flow diagrams for generating visualizations

a single color but a gradient of colors. Look at the first example, the length of period is 1 day, so the time bin for each column is 900 seconds. Following the data quality definition, there are two kinds of green in the first example, one representing the at least one data point and the other representing more than 5 data points. In the second example, the length of period changes to 2 days so that the time bin for each column is 1800 second which contains two scans from Location probe. Then, there are more kinds of data quality for a single time bin than the first example. The data qualities in the second example are as following:

- Both of two scans have more than 5 data points.
- One of the scan has more than 5 data points while the other scan has at least one data point but no more than 5.
- Both of two scans have at least one data point but no more than 5.
- One of the scan has at least one data point but no more than 5 while the other scan has no data point.
- Both of two scans have no data points.
- One of the scan is missing.

The measurement of data quality is based on the number of scans with one or more data points within the time bin. In the third example, there are more kinds of data qualities. By using this kind of color encoding, we can get not only the data quality is good or bad but also how good is the data. Then, it is possible to find out some participants with best data quality for research.

6.3.4 Interaction

In order to improve the visualization and user experience, several interaction techniques is applied in the system. For instance, Figure 6.16 shows the usage of the new feature filtering in the user matrix. The user is able to filter out the data from other probes and look at the data quality for only one probe. In the example, by comparing the participants' data quality from different probes, it is easy to find that the most of missing data occur at the Location probe which leads to a group of participants having missing data as the overall data quality. Furthermore, from the general view point, we can find the data quality of Bluetooth probe and Cell Tower probe is much better than the other two probes.

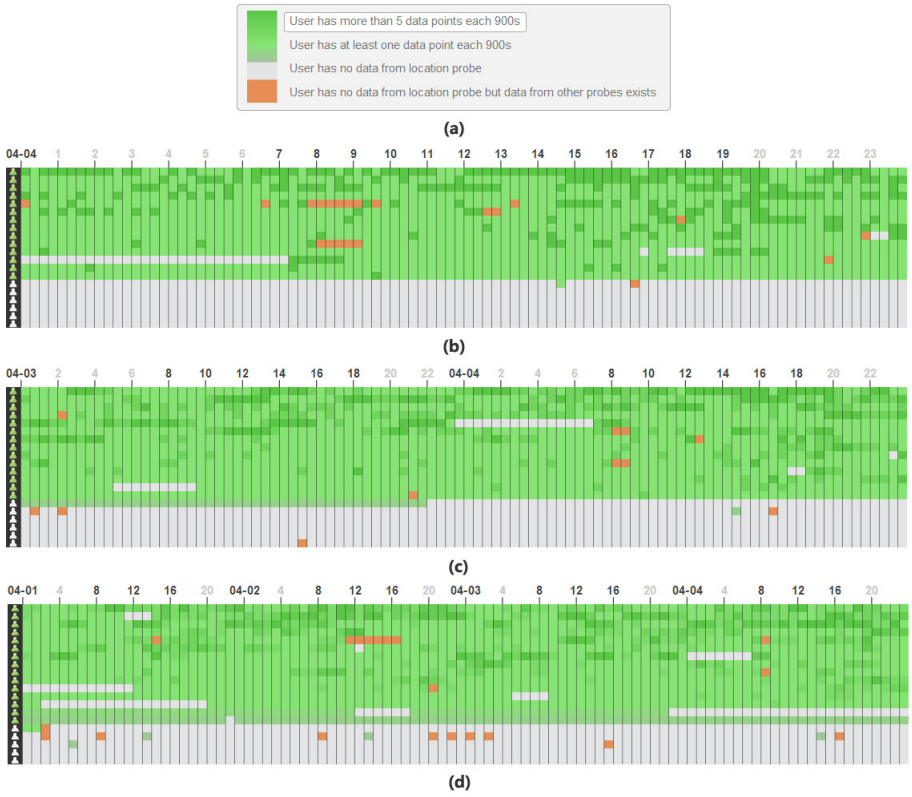


Figure 6.15: Examples of matrix visualization for Location probe. (a) The color encoding of data qualities. (b) Matrix visualization of 1 day period for Location probe. (c) Matrix visualization of 2 days period for Location probe. (d) Matrix visualization of 4 days period for Location probe.

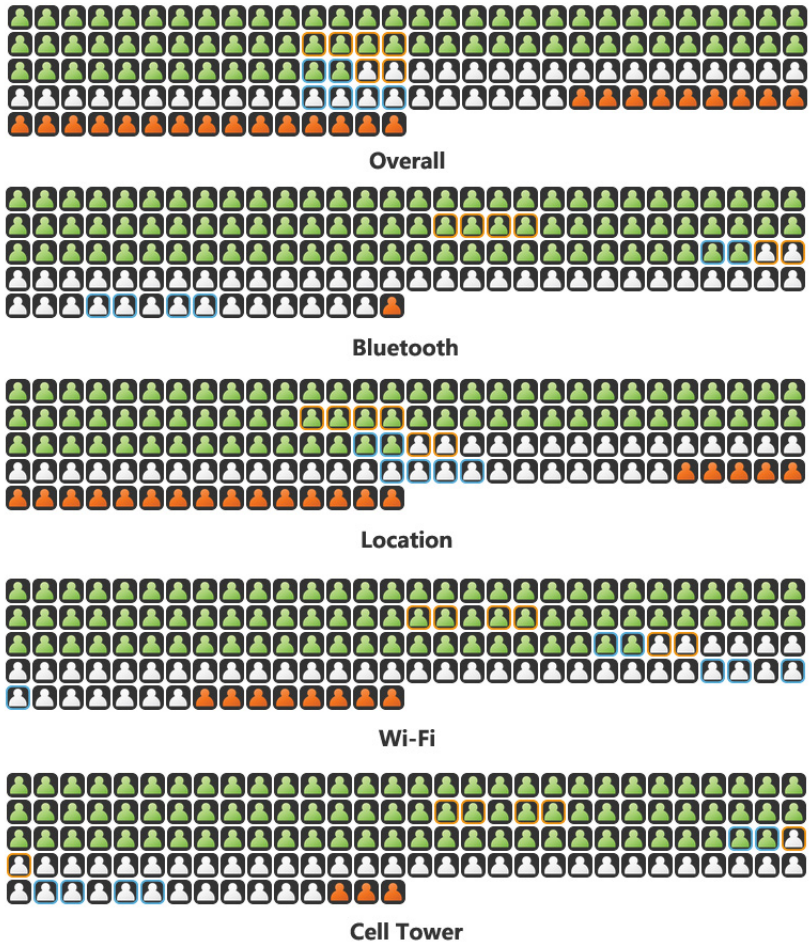


Figure 6.16: Filter technique applied in user matrix. After the filter applied, the user matrix only visualizes total data quality from the specific probe. The order of the matrix is rearrange by the total data quality of the specific probe.

The ordering operation applied on the visualizations is one kind of the reconfigure technique. In a big visualization, it is useful for finding out the information the user is interested in. When the filter operation is applied on the user matrix, the color of the icon is changed to the new view and then the order of participants is rearranged by animation as shown in Figure 6.17. After the rearrangement, it is hard to find where the participants added to the user list by the user are. So, the select technique is applied as the highlights in the matrix. The user can get track of the participants with the color of highlight even if the view is changed.

The matrix visualization is generated separately from the user matrix and user list. It is not a wise idea to let the user connect them manually by the ID of the participant. The connect technique is applied to solve the problem. As discuss the before, in the matrix visualization, there is a user icon in front of each row. The user can click on the icon, and then the system will jump back to the user list and select only the participant the user has clicked. Since the connect technique applied this way builds a link from the matrix visualization to the user list, this solution has some limitations, such as the user can get track of only one participant each time.

6.4 Backend

6.4.1 Data collector

The purpose of the data collector is to keep the system's data quality database up to date by updating the data quality from Sensible DTU database periodically. Due to the size of data for all participants is quite big, the update procedure is divided into small updates of one hour each time. For example, the daily update downloads all of participants' data from yesterday, in which the exact update procedure is downloading data of one hour size 24 times from the Sensible DTU API. By splitting the data downloaded for update, the cost of retry is reduced if an error occurred while downloading the data. After downloading the data, the system transforms the data into data quality information. One important step is binning the data. Since the sampling times from participants are random, we need to align them to the scan rate of the probes. Base on the scan rate of probes, the raw data is distributed into time bins. For each time bin, the start time is used as the timestamp of the time bin. Then, when the data quality information is recorded in the data quality data base, the timestamp of the raw data is changed to the time bin's timestamp. This preprocess method can reduce the work of generating the data for visualization.

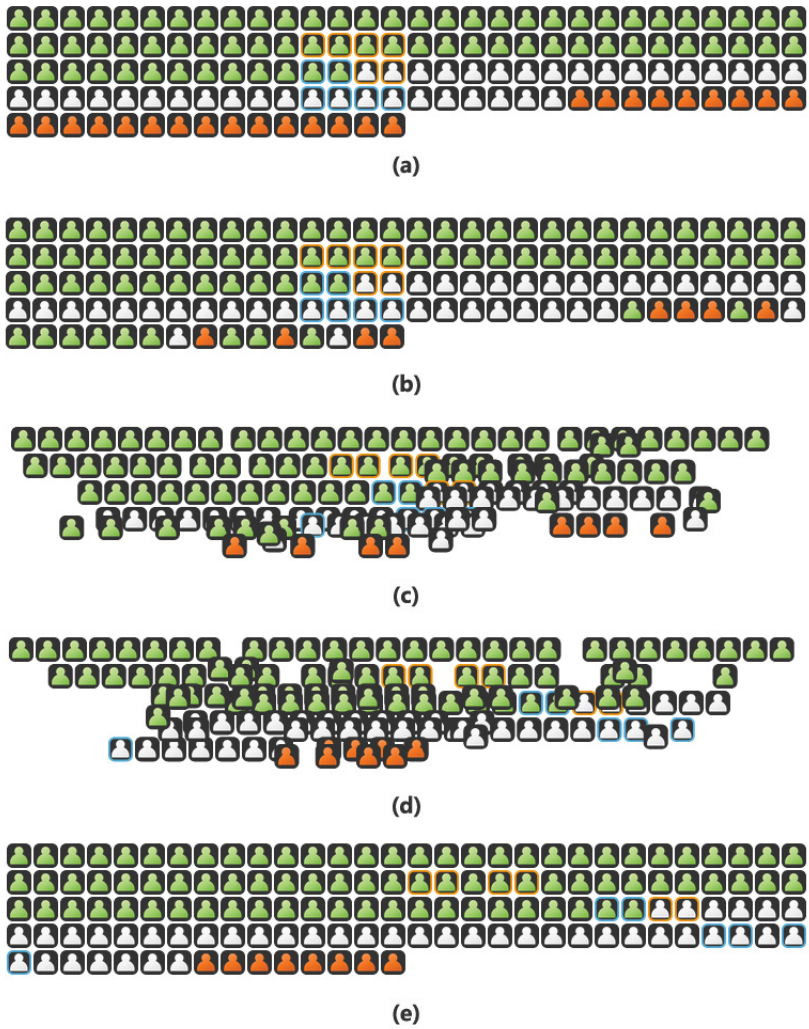


Figure 6.17: Reorder animation of the user matrix. (a) The user matrix of overall data quality. (b) After the filter is applied, the total data qualities of participants change to the specific probe. (c), (d) The user icons move to the new location by smooth transition. (e) The new arrangement of the user matrix based on the total data quality of the specific probe.

6.4.2 Data Query

In order to generate the visualization, the client needs to query the data from the server. While handling the query request from the client, the server first queries the data from the data quality database based on the period of time in the request. Then the server generates the data quality information in JSON format. The response of the server is the JSON format string with data quality information which can be used by D3.js directly.

While generating the data quality information, the server needs to compare data from all probes for checking the missing data. Then, the server queries the data from all positioning probes which limits the performance of the system. One possible solution is records the information of all time bins that have data from any probes. Then, if the visualization is for a single probe, the amount of the data needed to query is decreased by half. On the other hand, recording additional information will increase the size of data quality database and it does not improve the performance for generate overall data quality. Due to the space limitation of the test server, the solution is not applied to the system.

Chapter 7

Evaluation

7.1 Feedback

Since the system is not deployed to public, there are only a few of users participated in the test of the system. The amount of feedback is not enough for statistical analysis, but there are still some useful feedbacks for improving the system.

One important aspect of the feedback is that the system is useful if the user knows what to do with the system, but it is hard to understand all of the system if the user has no knowledge about the Sensible DTU project. Furthermore, the function of UI components is not easy enough to understand before several trials. One suggestion for improvement is to add a quick tutorial of how the system works in the front page, which gives more information about Sensible DTU project and the objectives of the system.

One user is not satisfied with the loading time of the visualization. When loading the visualization for one month, the user has to wait about one minute. Even if there is loading animation indicating the system is working, the long loading time still decrease the user experience. The suggestion is adding the information of loading progress that can give user an approximate time of when the visualization will be generated.

In addition, one user considered to use the system on a smartphone which is nearly impossible with the current system. The smartphone has become one of the mean platforms for using the web application in recent years. During the design of the system, only the capability of browser for desktop computers is took into consideration. The purposes of the system is to provide an easy approach for checking the data quality in the database, so making the system available for smartphones is an important improvement in the future work.

7.2 Performance

The performance of the web application plays an important role in the user experience. The loading time of the visualization is a b. A series of performance tests were done in the environment of a laptop with a 2 Ghz Quad-core Intel Core i7 processor and 4 GB RAM. The running time of generating different visualizations are shown in Table 7.1. Each test was run 20 times and the average value was taken. Since the running time for rendering the visualization in the browser is normally less than one second, the most of loading is for the server to query the data quality information. From the result, the runtime of loading the visualization is linear based on the length of period. Due to the amount of data needed to query for generating the overview for overall data quality is the same as the overview for Bluetooth probe, the loading time of two different probe are similar.

Lenghth of period	Overview of overall data quality	Overview of Bluetooth probe	Overall matrix visualization of 50 participants
1 day	2.5 s	2.1 s	2.4 s
2 days	4.2 s	4.0 s	3.8 s
4 days	7.0 s	6.2 s	5.5 s
7 days	11.2 s	10.8 s	7.6 s
14 days	19.4 s	20.0 s	12.6 s
30 days	40.5 s	42.2 s	25.4 s

Table 7.1: Loading time for generating different kind of visualizations

On the other side, the running time for updating the data quality database is another factor of performance. Since the time of downloading the raw data from Sensible DTU database API is affected by internet connection, we only consider the running time of processing the raw data. The performance test for update was done in the same way as the previous work. The result of the test is shown in Table 7.2. Two updates of different time are chosen for test, both of which

are one hour period. From the result, the running times for the two updates are nearly the same. Table 7.3 shows the size of the raw data from the Sensible DTU database for the two updates, it is clear that the second update has much more raw data than the first. Furthermore, the update for Bluetooth probe takes the most of runtime in update while the size of raw data from Bluetooth probe is much less than the raw data from Wi-Fi probe. The reason is that the Bluetooth probe has the highest scan rate in the positioning probes so that there are more records from Bluetooth probe then from other probes in the data quality database. As the consequence, the factor that has most affected on the running of update is the number of write operations to the data quality database. Due to the update process runs periodically on the server, the running time of the update will not put a high workload on the server.

Probe	04-15 00:00 to 01:00	04-15 12:00 to 13:00
All	51.8 s	53.4 s
Bluetooth	25.1 s	24.6 s
Location	9.5 s	10.2 s
Wi-Fi	15.0 s	17.8 s
Cell	0.8 s	0.9 s

Table 7.2: Processing time for updating the data quality database.

Probe	04-15 00:00 to 01:00	04-15 12:00 to 13:00
Bluetooth	290 KB	1.2 MB
Location	1.5 MB	2 MB
Wi-Fi	19.3 MB	36 MB
Cell	85 KB	83 KB

Table 7.3: Size of raw data from Sensible DTU database.

As the system maintenance a data quality database separately from the Sensible DTU database, the size of the data quality database is one factor of the performance. Generally, the size of raw data for one day from the Sensible DTU database is about 800MB that consists of 15MB data from Bluetooth probe, 70MB data from Location probe, 700MB data from Wi-Fi probe and 3MB data from Cell Tower probe. The size of the data quality database for one day is about 4MB that is 5% of the size of raw data, which is available to be maintained by a lightweight database. Since there are only 135 participants in the Sensible DTU database, if the number of participants increased to more than 1000, the size of the data quality database should be took into consideration for the future works.

Chapter 8

Discussion

From the feedback, the Sensible DTU Data Monitor is a working data reporting system for the Sensible DTU database. There are some good features in the system while the system is a start point of data monitoring in Sensible DTU with plenty of possible improvement.

Generally, when users get some trials on the system, they can easily understand the visualizations and identify the participants with high quality data or low quality data. The different level of abstraction of data quality information in the visualizations is useful when the user want to find some specific information from a long period of time. By applying several visualization interaction techniques discussed in Chapter 2.3, the user experience is improved. Furthermore, the system first gives users the view from the top level of database working status, by interacting with the visualization, users can dig into the data quality information with highest resolution.

For the future work, there are a lot of possible improvements for the system. As basic improvements, the performance of the system is limited by the database which is necessarily to be improved. One solution is to record additional data quality information during the update for saving the time of generating the visualizations. The improvement of this solution needs further test. In the Sensible DTU project, there will be a new API for the database which provides better performance and more operations on the database level. Base on the

new database API, there is possibility to do all of the work for data quality on database level and skip the maintenance of the separate data quality database.

The current version of the system works only with the positioning probes, while there are more probes in the Sensible DTU project. Due to the randomness of the social probes, the matrix visualization is not a wise choice. If using the matrix visualization for social probes, there will be a big matrix with most area filled with empty data. In order to make the system available for all probes in the Sensible DTU project, other kind of visualization is needed.

Besides the data quality information, it is possible to integrating more information into the visualization as the related work discuss in Chapter 2.2. In order to integrating more information, the raw data is needed for each query. In the current system structure, it is impossible to download the raw data for each query. With the support of the new database API, it may be possible to get the processed data from the database instead of downloading the raw data.

Chapter 9

Conclusion

This thesis is a case study of designing and implementing a data reporting system in real time that gives the user an easy approach for checking the data quality in the database. The system is based on Sensible DTU project which has a big data collection of participants' digital traces in their daily lives.

From the evaluation and discussion in previous chapter, the work of this thesis meets the goal that the data reporting system can visualize the data quality of Sensible DTU database in different aspect with different level of detail. Furthermore, the people working in the project can get benefit from the data reporting system. For database maintainers, they can easily check if the database is working properly and find out participants whose data collector is not working properly. For researchers, they can find out a group of participants with high quality data for their studies.

The system implemented in this thesis is the start point of data monitoring of the Sensible DTU project, there are still a lot of work to do to improve the system. The future work includes improving the performance of the system, visualizing the data quality for all probes and integrating more information other than data quality for the data reporting.

Bibliography

[1] David Lazer, Alex Sandy Pentland, Lada Adamic, Sinan Aral, Albert Laszlo Barabasi, Devon Brewer, Nicholas Christakis, Noshir Contractor, James Fowler, Myron Gutmann, et al. Life in the network: the coming age of computational social science. *Science (New York, NY)*, 323(5915):721, 2009.

[2] Sensible dtu high resolution social networks, http://www.sensible.dtu.dk/.

[3] Leo L Pipino, Yang W Lee, and Richard Y Wang. Data quality assessment. *Communications of the ACM*, 45(4):211–218, 2002.

[4] Yair Wand and Richard Y Wang. Anchoring data quality dimensions in ontological foundations. *Communications of the ACM*, 39(11):86–95, 1996.

[5] Gustavo EAPA Batista and Maria Carolina Monard. An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, 17(5-6):519–533, 2003.

[6] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*, volume 539. Wiley New York, 1987.

[7] Barry N Taylor. *Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results (rev.* DIANE Publishing, 2009.

[8] Matthew O Ward. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization*, 1(3-4):194–210, 2002.

[9] Tamara Munzner. *Interactive visualization of large graphs and networks.* PhD thesis, Citeseer, 2000.

[10] Shiping Huang. *Exploratory visualization of data with variable quality.* PhD thesis, WORCESTER POLYTECHNIC INSTITUTE, 2005.

[11] William S Cleveland. *Visualizing data.* Hobart Press, 1993.

[12] Toby Segaran and Jeff Hammerbacher. *Beautiful data: the stories behind elegant data solutions.* O'reilly, 2009.

[13] Julie Steele and Noah Iliinsky. *Beautiful visualization.* O'Reilly Media, Inc., 2010.

[14] Zaixian Xie, Shiping Huang, Matthew O Ward, and Elke A Rundensteiner. Exploratory visualization of multivariate data with variable quality. In *Visual Analytics Science And Technology, 2006 IEEE Symposium On*, pages 183–190. IEEE, 2006.

[15] Ji Soo Yi, Youn ah Kang, John T Stasko, and Julie A Jacko. Toward a deeper understanding of the role of interaction in information visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1224–1231, 2007.

[16] Introducing json, http://www.json.org/.

[17] Trulia trends, http://trends.truliablog.com/2011/09/house-hunter-by-day-not-so-much-after-midnight/.

[18] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D^3 data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2301–2309, 2011.

[19] Processing.js, http://processingjs.org/.

[20] jquery, http://jquery.com/.

[21] jquery ui, http://jqueryui.com/.

[22] Google app engine, https://appengine.google.com/.

[23] Django, https://www.djangoproject.com/.