@HBS >An app-assisted approach for the Household Budget Survey

Report on the action @HBS

Version 1

Barry Schouten (Statistics Netherlands, jg.schouten@cbs.nl)
Joanna Bulman (UK Office for National Statistics)
Marjaana Järvensivu (Statistics Finland)
Marc Plate (Statistics Austria)
Brigita Vrabič-Kek (Statistics Slovenia)

project number 847257

February 28, 2020

summary This report on the action is the main deliverable of project @HBS. It describes the main activities and results of the @HBS consortium of ONS, Stat Austria, Stat Finland, Stat Netherlands and Stat Slovenia in the three topical work packages. It also provides a guide to all detailed @HBS reports. @HBS investigates an app-based approach to the Household Budget Survey.

ACKNOWLEDGEMENTS: To this paper many others have contributed: Annette Jäckle (university of Essex), Anna Niemelä (Stat Finland), Deirdre Giessen, Jelmer de Groot, Jack Janssen, Mart Nyholt, Tom Oerlemans, Stefan Theunissen, Nick de Wolf (Stat Netherlands), Mateja Zgonec (Stat Slovenia) and Joanna Bulman, Rob Doherty, Lanthao Benedikt, Louisa Nolan (ONS).

Contents

Summary 3

1. Introduction to the @HBS project 3

2. The Household Budget Survey app 5

    2.1 Frontend 5

    2.2 Backend 7

3. Evolution of the @HBS app through two test rounds 7

4. Motivation and recruitment strategies 8

5. Smart features 10

    5.1 Receipt scanning 10

    5.2 Big data linkage 12

    5.3 Geo-locations 14

6. Future development 16

Appendix A – Overview of @HBS supplemental products 18

Appendix B – Backend specifications 19

Appendix C – Screenshots HBS app 24

Appendix D – Screenshots HBS app bank mockups 28

Summary

Main conclusions from @HBS:

- An app-assisted approach is received very well by test persons in all three countries
- Test persons view an HBS app not just as an official statistics tool, but also as a personal tool; expenditure statistics are perceived as useful
- The most complex features in an HBS app are product search lists and reporting prices for products with certain types of discounts (e.g. take three, buy two)
- Country-specific features of a basic HBS app are (apart from NSI logo and app name) user interface language, product search lists, store search lists, helpdesk contact information and the type of feedback on incentives earned
- Country-specific features of a receipt scanning HBS app are receipt format and structure, and the availability of GTIN/EAN product descriptions for matching receipt text
- Product search lists are time-consuming to prepare and need constant updating
- Tests suggest that respondent recruitment and instruction are crucial, but, once recruited, respondents remain motivated and perceive the respondent burden as relatively low
- Consent questions for linkage to loyalty card data, scanner data and bank transactions data require a careful design of the UI; adding these additional data sources to the app may make reporting expenses more complicated and may have relatively limited added value as respondents see entering expenses in the app by themselves as rather easy.
- Willingness to consent to linkage to bank transactions data demands very convincing arguments that linkage removes respondent burden and is safe
- Linkage to loyalty card data and/or scanner data was perceived as a more natural option, but still demands convincing arguments
- Geo-locations were perceived as hardly useful and intrusive in the context of the HBS

Main recommendations from @HBS:

- Harmonize product search lists across ESS countries
- Develop a full receipt scan processing pipeline with instant feedback of classified receipts to respondents
- Devote extra care and effort to material for recruiting and instructing respondents
- Further explore consent to link bank transactions data and the utility of such data, especially in designing the UI such that it supports and not complicates the response task
- Further explore consent to link scanner data as a precursor to digital receipts

1. Introduction to the @HBS project

Within the ESTAT call Modernization of HBS and TUS, @HBS has developed and explored an app-assisted approach towards data collection for the household budget survey (HBS). In particular, it explored three sensor data extensions to reduce the respondent burden. As a starting point and to focus on the sensor data extensions, the app is modelled after the existing HBS app developed by Stat Austria. This app does not employ sensors but facilitates keeping a budget diary by a household. The three extra features that are investigated are receipt scanning and classification, geo-locations, and big (sensor) data linkage through consent. The receipt scanning is performed in-app for the infrequent expenditures concerning multiple products. The geo-locations are based on location measurements and are a form of geo-fencing where the app detects a close vicinity to shops and other sites where expenses are likely. The big data linkage concerns the necessary extra information to uniquely link respondent expenditures to relevant data such as scanner data or bank transactions data.

@HBS was a consortium of ONS, Stat Austria, Stat Finland, Stat Netherlands and Stat Slovenia. Stat Netherlands was the coordinator and was responsible for the development of the @HBS app. The project had three topical work packages which comprised of the following tasks:

- Work package 2 (coordinated by NL) – app development:
    1) Copy main features of the Stat Austria app
    2) Separate generic and country-specific components
    3) Implement receipt scanning
    4) Implement geo-locations
    5) Implement big data linkage and consent examples
    6) Describe specifications for the app backend
- Work package 3 (coordinated by NL and SI) – usability testing and data collection strategies:
    1) Test the basic app in two rounds in three countries
    2) Test and report each of the three extensions
    3) Recommend candidate recruitment and motivation strategies
- Work package 4 (coordinated by NL and UK) – data analysis and data linkage:
    1) Make an inventory of potentially relevant big (sensor) data
    2) Identify necessary information to link big (sensor) data after consent
    3) Explore OCR and classification of scanned receipts
    4) Validate, if feasible, in-app scanning with in-house scanning of receipts
    5) Implement instructions for in-app scanning based on the in-house and in-app comparison

Appendix A provides an overview of all sub-deliverables in the three work packages.

Out of the three smart features, receipt scanning was implemented and has been thoroughly tested, big data linkage has been explored, tested but not implemented, and geo-location has been explored but not tested or implemented. The motivations will be given in detail in this report in section 5. In sum, after a number of brainstorms and based on test person comments, the international project team rated

- the receipt scanning as very promising,
- the big data linkage as promising but associated with a number of complex methodological and UI questions, and
- geo-locations as only mildly useful for respondents, except perhaps when combined with big data linkage.

Although it is not fully implemented, @HBS has gained very useful knowledge on big data linkage options, which will make it relatively straightforward to design and initiate follow-up research and development.

In the following, it is assumed that the reader is familiar with the HBS data collection and classification. In section 2, we present frontend and backend of version 3 of the @HBS app.

In section 3, we summarize the evolution of the app through two rounds of testing in three countries. In section 4, we elaborate on recruitment and motivation strategies. In section 5, we discuss each of the three smart features. We end with future development in section 6.

2. The Household Budget Survey app

The @HBS app is listed as Household Budget Survey app in iOS and Google app stores. The source code can be found at

https://gitlab.com/tabi/hbs-budget-app/budget-app-cbs

The app will be submitted to the Eurostat inventory of tools following CSPA (Common Statistical Production Architecture) guidelines. The documentation can be found at:

https://webgate.ec.europa.eu/fpfis/wikis/display/ISTLCS/INVENTORY+of+Tools+and+Sources+for+HBS+and+TUS

The Household Budget Survey app is at version 3 at the time of writing. Here, is the description of versions:

- Version 0: Version prepared to evaluate technical problems across countries and devices (June/July 2019)
- Version 1: Version with revisions of technical issues for test round 1 (August 2019)
- Version 2: Version with revisions based on test round 1 (December 2019)
- Version 3: Version with revisions based on test round 2 (March 2020)
- Version 4: Version including receipt scanning pipeline (expected June 2020)

In this section, we describe the frontend and backend of version 3 of the Household Budget Survey app. In doing so, we distinguish generic and country-specific elements.

2.1 Frontend

Currently, the Household Budget Survey app is available in Dutch, English, Finnish and Slovenian. The English version is the international demo version of the app. To avoid extra work, one app was submitted to app stores and respondents need to choose the language. This can easily be changed. In preparing country apps, the language (not the content) can also be based on the language settings of the smart device.

Generic features of the app frontend are:

- Diary functionality with option to pre-specify the data collection period
- Manual data entry and receipt scan data entry
- The use of separate product search lists linked to COICOP
- The use of separate store lists to prepare receipt processing and classification of products
- Four main screens: calendar, list of submitted expenditures, expenditure statistics and settings
- Check of receipt scan data quality (UNDER DEVELOPMENT FOR VERSION 4)

Country-specific features of the app frontend are:

- App name and colour scheme

- App language
- Content of product search lists
- Content of store search lists
- Helpdesk functionality (helpdesk phone number and email)
- Display of incentives earned by the respondent

The country-specific features of the app mean some time investment for each NSI that would like to employ the app for their HBS. What is needed?:

- App texts need to be translated: This task by itself takes little time as the amount of text used is limited. However, some iterations may be needed to find the right balance between length of texts and navigation/buttons. For this purpose, within @HBS, a file has been prepared with all app texts in English next to screenshots
- Product search lists need to be prepared and linked to COICOP. In @HBS, FIN, SLO and UK did not have extensive lists, but only smaller lists designed for manual coding. The preparation of lists can be time consuming. The NL list contains 29.000 entries. Within @HBS, a simple web scraping program was developed that translates NL product lists to any specified language and that may be used as a starting point
- Store search lists need to be prepared. Store lists are uncommon to most NSI's and demand some time investment. The stores are used for three purposes: To assist processing of scanned receipts, to pre-classify products into main categories for scanned receipts, and to prepare/anticipate digital receipts in the near future
- Helpdesk contact information needs to be added. This task, obviously, implies much more than just a phone number and email. We will return to this in the backend section.
- The app needs to be submitted to app stores by the NSI. This task requires app store accounts.

A number of design choices has been made in the app:

- Multiple household members can use the same app, but expenditures are not anonymized within the household
- Product search for manual data entry is based on two principles: At the start of data entry only recently bought items are shown. When typing, products are suggested based on a distance metric
- Store search for all data entry is based on the same principles as for products
- Respondents get instant feedback in two ways: a circle diagram with total amounts and amounts per first digit COICOP, and a timeline with expenditures per day split over first digit COICOP

The best way to explore and experience the Household Budget Survey app, is to download the app and to use it for one or a few days. Here, we merely provide screenshots of the most important screens.

Test logins for the app can be requested at jg.schouten@cbs.nl .
The app is available as Household Budget Survey in Google and Apple stores.

Screenshots of the app can be found in Appendix C.

2.2 Backend

Currently, the Household Budget Survey app is linked to a basic backend that supports data submission, but that does not yet have full monitoring functionality and that is disconnected from receipt processing. In the two test rounds, the backend was not linked and no data was transmitted. Instead a separate app was developed for the submission of receipts. The resulting database of receipts has been used to explore machine learning models for receipt text recognition.

The backend is implemented as a GO-server with a PostgreSQL database. The backend has four main functions:

- User authentication: check whether the user password matches the (hashed) password in the database
- Store receipt data: the data gathered on the smartphone is also stored in the database.
- Synchronize receipt data: keep the data synchronized between smartphones if the user uses multiple smartphones.
- Monitoring: respondent activity and app usage is monitored and may be input to follow-up actions

Generic features of the app backend are:

- Uniform data model
- Transmission of data when device is connected to Wi-Fi
- Monitoring of survey data, receipt data and paradata (app activity and navigation) through queries on the database (TO BE DEVELOPED FOR VERSION 4)
- In-house receipt scan text recognition (TO BE DEVELOPED FOR VERSION 4)

Country-specific features of the app backend are:

- In-house classification of receipt texts (TO BE DEVELOPED FOR VERSION 4)
- Incorporation into case management systems and strategies, including helpdesk functionality

Appendix B provides specifications for the backend.

3. Evolution of the HBS app through two test rounds

The Household Budget Survey app went through three versions during project @HBS. In the coming two years, the app frontend and backend will get major revisions because of the inclusion of the receipt processing pipeline.

The two test rounds' protocols, plans and findings are described in great detail in separate sub-deliverables. In all cases, the app was tested in Finland, the Netherlands and Slovenia. Desk reviews of the app have been given by ONS, by University of Essex and by Statistics Canada. Recommendations from these desk reviews have been included as well in the app user interface. The English version of the app is used for demo purposes and went through the same revisions as the Dutch, Finnish and Slovenian versions.

In test round 2, consent to link to loyalty card data, scanner data and bank transactions data was evaluated in the Netherlands only through mock-up screens. The extra screens were not included in the app itself, but were simulated on a smartphone. The mock-ups do not support actual data entry, but are realistic in all other aspects. In addition, test persons received a paper instruction how to add bank accounts. A separate sub-deliverable is prepared including screenshots of the consent linkage screens.

What has changed from version 1 (August 2019) to version 3 (March 2020)?

These are the main changes implemented after test round 1:

- Adding a walk through tutorial for each screen
- Adding tips for making a picture of a receipt
- Improved design for the screen for manual entering of expenses; the flow of this screen was one of the main issues found in test round 1
- Adding the option to enter negative expenses (e.g. for refunds)
- Enabling entering expenses directly via calendar
- Adding more safety to prevent accidental editing/deleting of expenses
- Several technical issues / bugs

These are the main changes implemented after test round 2:

- Improve usability of walk through tutorial
- Improve explanation of how to report prices with a discount
- Several technical issues / bugs (e.g. not consistently displaying expenses entered with a picture in overview screen)

4. Motivation and recruitment strategies

During project @HBS, only small-scale, qualitative tests have been conducted, but data collection strategies have been explicitly included in the two test rounds. Each test round consisted of two groups: a full observation group and a retrospective group. The retrospective group was invited to register and open the app themselves with just the help of the recruitment material. The full observation group was also monitored while performing this task.

We distinguish recruitment and motivation strategies. Recruitment activities take place before the actual data collection and motivation activities during data collection. A literature study revealed relevant papers by the university of Essex (Wenz, Jäckle and Couper 2019 Survey Research Methods, Read 2019 Survey Research Methods, Jäckle, Burton, Couper and Lessof 2019 Survey Research Methods). The number of reported app-assisted HBS-type surveys is still relatively low, however. University of Essex acted as advisor on @HBS.

What are design features in recruitment:

- Type and height of incentives
- Employment of interviewers
- Content of invitation and reminder letters
- Use of an information leaflet
- Use of an installation guide
- Use of an online landing page

- Provision of individual household statistics about expenditures

What are design features in motivation:

- User interface
- Conditional incentives
- Contacts by an interviewer
- Plausibility checks
- Provision of individual household statistics about expenditures
- Helpdesk (phone, email, social media) for technical support
- In-app notifications
- Push-notifications
- Other household contact info (email, phone number)

In qualitative tests, invitation letters, leaflets, installation guide, and online landing page could be evaluated for recruitment purposes and user interface, plausibility checks, in-app notifications and helpdesk for motivation purposes. The provision of household expenditure statistics can be evaluated for both recruitment and motivation purposes.

We did not evaluate all design features. In the two test rounds, the option of a landing page was not used. Test persons could make use of the in-app notification functionality, but this was not explicitly stressed. Plausibility checks were only implemented in a rudimentary form; test persons have to validate each calendar day as completed. The other design features were evaluated. In supplemental material, an invitation letter, a flyer and an installation guide are provided. The user interface, obviously, was the most important element of the test rounds.

Table 4.1 provides an overview of the current recruitment and motivation strategies in the HBS for the five @HBS countries. A prominent feature in all countries, except NL, is the employment of face-to-face interviewers. Interviewers may have four roles: recruit sample households at home and/or over the phone, conduct a basic recruitment survey, assist in installing and operating an app, and motivate during the data collection period. In the usability tests, only the third role of assistance could be evaluated.

Table 4.1: Recruitment and motivation strategy choices in the @HBS countries.

	AT	FI	NL	SI	UK
Reporting period	2 weeks in the diary, questionnaire up to 12 months	varying	4 weeks	Varying	2 weeks
All expenditures	2 weeks for all	2 weeks for all	1 week all 3 weeks >20 €	2 weeks for all	Yes (see separate list)
Interviewers recruitment	Yes, F2F; additionally, without interviewer, for CAWI-only	Yes, interviewers choose F2F or phone	Yes, only if phone nr available	Yes	Yes, F2F

Interviewers motivation	Yes, by telephone is recommended to interviewers but not mandatory	Yes	Yes, during 2nd week	Not during fieldwork	Yes, checking calls
Unconditional incentives	Shopping bag with HBS logo	Small gift (1 €)	5 €	No	2nd class Stamps
Conditional incentives	50€ voucher for shops in Austria	Lottery (2016) 10-20€ (2012)	30 €	No	£20
Info or landing page	Info page Yes, landing page No	Yes	Yes	No	Yes

The two test rounds revealed, especially, the importance of clear instruction material, both on paper and in-app. We learned that it must be assumed that respondents will only partly read instruction material, as even test persons did not while instructed to. This finding implies that in-app help and an intuitive, familiar UI are imperative. Based on the first test round, the in-app tutorial was revised completely and made very prominent. The consequence of all this is that landing pages are less important as a tool to provide instructions and answers to frequently-asked-questions.

5. Smart features

We describe the three main smart features that have been explored in @HBS: receipt scanning, big data linkage and geo-locations.
The Python source for receipt scan text recognition is available at

https://gitlab.com/Nickdewolf/hbs.wp4-shopping-receipt-recognition

The big data consent mockups in Figma format are presented in Appendix D but are also available at

https://www.figma.com/proto/x270TIsAatqM7uZtATP3cQ/Bank-koppeling-proces-prototype-HBS-englisch?node-id=3%3A684&scaling=min-zoom

5.1 Receipt scanning

Most of the research and development time in @HBS on smart features has been devoted to data entry through receipt scans. The focus was on the utility and usability of the inclusion of receipt scanning in the app user interface for respondents. Although, it was not formally part of @HBS, also a receipt scan processing pipeline has been investigated.

The two test rounds clearly revealed the added value of receipt scans as a data entry option to remove respondent burden. The test persons considered the data entry option to be useful and relatively easy to operate. In the first test round, no special instructions were given, other than stressing that the receipt is fully scanned. In the second round, test persons did receive in-app instructions about scanning. These instructions were based on OCR machine learning models‘ performance. The test persons were instructed:

- To photograph receipts in daylight
- To avoid white backgrounds when photographing
- To photograph receipts straight from above
- To avoid shadows
- To make sure that receipt edges are straight and visible
- To avoid curves in the receipt

These instructions were only shown the first time a scan is made. They could be displayed for later scans by clicking on the information button.

In @HBS, the assumption was that receipts are scanned in-app. However, various countries ask respondent to collect receipts and send them to a specified postal address where the receipts are scanned in-house. This is an expensive option that also complicates (instant) feedback on expenditures to respondents. Within WP4 of @HBS, a first evaluation was made of accuracy of in-app versus in-house scanning. In-house scanning is performed using a flatbed scanner. The number of receipts in the evaluation was small. Results showed that the accuracy between in-app and in-house scanning was relatively similar if test persons follow in-app instructions. The number of receipts was too small to draw string conclusions, but these results are promising.

The full receipt scan processing pipeline is anticipated to consist of nine steps:

1. Respondent photographs receipt (in-app)
2. Text recognition is applied and instructions are given to respondent (in-app)
3. Scan is sent to web server (in-app)
4. Scan is transformed to symbols (in-house)
5. Symbols are transformed to text (in-house)
6. Product-price pairs are derived from text (in-house)
7. Products are classified to COICOP (in-house)
8. Classified products-prices are returned to respondent (in-house)
9. Respondent checks classification (in-app)

Step 2 may be seen as a preliminary version of steps 4 to 6. As much as possible may be done in-app in order to avoid data exchange between respondents and backend. Steps 4 to 6 may become obsolete, when text recognition accuracy is sufficient. However, accuracy may, in general, be too low to avoid in-house text recognition. In order to avoid low quality scans, an in-app first check is imperative and step 2 provides feedback and instructions to respondents.

Step 7 is specific to HBS. There are two settings. The first setting is where GTIN/EAN product descriptions are available at the national statistical institute as a byproduct of scanner data. Typically, these descriptions are linked to COICOP categories on an on-going, continuous basis. Classification of products comes down to matching in this setting. The second setting is where suck descriptions are absent and classification needs to be based on machine learning models that are trained with a large set of annotated and coded receipts. In @HBS, Stat Finland, Stat Netherlands and Stat Slovenia are under the first setting and ONS is under the second setting.

It is very likely that step 7 will need to be done in-house as products are too dynamic to use fixed in-app lists of product descriptions and/or to fix in-app machine learning models. Since part of the processing is outside the app anyway, it remains an option to perform part of the text recognition.

The @HBS WP4 report by Benedikt, Joshi, Nolan, De Wolf and Schouten gives a detailed account of the various steps as well as the performance and accuracy. We strongly recommend to read this report for those interested in receipt processing. The report forms the basis for on-going development of the receipt scan processing pipeline that currently takes place outside @HBS. In Section 5 of the report, a first comparison is made between in-app scans and in-house scans. Surprisingly, the accuracy and percentages missed lines and extra lines, see table 5 in the report, for in-app scans are close to in-house scans. The number of receipts included is, however, yet too small to draw strong conclusions, but these results are promising.

5.2 Big data linkage

The second smart feature, big data linkage, has two main types:

1. Ask respondents for consent to link data already in the possession of the NSI
2. Ask respondents to themselves request data from holders of relevant data

Under the first type, the app asks for consent to link data at the NSI and may ask the respondent for additional information to perform the linkage. The consent may be retracted by the respondent, after which the app has to send a request to undo the data linkage. Given that individual household expenditure statistics are deemed an important incentive in recruiting and motivating respondents, the linked data needs to be sent and presented to the respondent in the app. The app may ask the respondent to check/validate the data. However, such respondent checks may introduce measurement errors themselves and may, therefore, be discouraged.

Under the second type, sometimes termed data donation, respondents allow the app to communicate with external data holder databases. The app sends the request for data, receives the external data, presents the data to the respondent, may ask for checks/validation of the data, and asks the respondent to submit the data. For this type, the NSI may only receive a summary or sufficient derived statistic.

Four relevant data sets have been identified before and during @HBS:

1. Barcode product descriptions and prices: In certain COICOP categories, products are linked to GTIN/EAN codes that are displayed as barcodes on products. Online databases exist that provide detailed descriptions of these products. Online databases may also exist that include prices of the products at different stores. Respondents may scan barcodes in-app and the resulting GTIN/EAN are linked to databases in-house.
2. Scanner data: Part of the NSI's in ESS countries receive scanner transactions data from one or more stores. For example, in NL around 30 stores deliver weekly aggregated scanner data. Individual scanner data transactions are usually not received by NSI's, but are available in store databases. Respondents would need to provide some information for linkage, either a receipt number or a mix of total expenditure amount, time and location. Based on the information, individual

transactions may be earmarked and sent. It must be noted, however, that such earmarking has to be done separately for each store and demands for extra effort by the stores. This type of linkage shows resemblance to on-going innovation at Eurostat involving iCards and/or dedicated apps with NFC (Near Field Communication) to store scanners.

3. Bank transactions data: Under PSD2 (Payment Service Directive 2, November 2015) legislation that applies to the EU, banks are obliged to provide API's that allow for communication with bank account holders. In order to access such transactions data, the respondent has to select banks, has to provide bank accounts and has to enable the functionality in-app to send requests to bank API's. Bank transactions data have limited metadata and respondents will have to supplement with extra contextual information. The combined data then is submitted.
4. Loyalty card data: In many countries, part of the stores employ loyalty cards to attract customers. By GDPR legislation, loyalty card holders can at any time request the data that is stored about them. The respondent has to select stores, has to provide loyalty card numbers and has to enable the functionality in-app to send requests to store API's or websites. The received data can be merged into the app and needs to be submitted

The first two big data sets are of the first type, the NSI performs the linkage, whereas the latter two are of the second type, the respondent requests the data about her/him.

The four options have been discussed and rated within project @HBS. With each option development costs and time are associated and processing and feeding back of data may be complicated. These need to be outweighed by removal of respondent burden and by the general willingness of respondents to really enable and employ the feature. Table 5.1 sets scores to each option on a scale. It must be added the scores are subjective.

Table 5.1: Scores of the four big data options on reduction in respondent burden, willingness to consent, complexity of app frontend and complexity of app backend. Burden and willingness are rated on a three-point scale of + to +++ with +++ the largest reduction of burden. Frontend and backend complexity are rated on a three-point scale from – to -– with --- the most demanding in development and maintenance.

	Reduction in respondent burden	Respondent willingness to consent	Complexity of app frontend	Complexity of app backend
Barcode data	+	++	-	-
Scanner data	+++	++	-	---
Bank transactions	+++	+	--	--
Loyalty card data	++	+++	--	-

Barcode data implies that respondents need to scan barcodes of all products with the mobile device on which the HBS app is installed and that the NSI maintains a database with all product descriptions linked to COICOP and prices. Implementation of in-app barcode scanning is relatively straightforward, although accuracy will depend on the device camera and the scan circumstances (light, contrast, folds). Scanning all product barcodes still poses some burden to the respondent. However, we expect that respondents will have few objections against scanning. The added complexity of the frontend is relatively small, but the maintenance and processing of the GTIN/EAN data plus prices demands for a complex backend, especially because the data need to be sent to the respondent.

Scanner data have become a standard source for many NSI's, but only at the aggregated level. In @HBS test rounds, test persons had few objections against linkage. To search for the transaction, respondents would, however, have to provide some information. One option may be to provide date, time and amount as presented at the receipt. Another, more intrusive, option would be to ask for location-time measurements. Scanner data does remove burden, but not completely. The app frontend would be easy to adapt. The most problematic is the backend as the NSI would have to negotiate the provision of individual transactions from all stores. Since it will cost extra effort and there is little added value, stores may be reluctant to do this.

Bank transactions data are to be linked as a form of data donation by the respondent. In @HBS test rounds, respondents were very mixed about the consent. The main reason was that they did not see the added value, i.e. how the NSI would transform the transaction data to statistics. This is a legitimate concern as transactions will only present total amounts of expenditures and metadata about the transaction is limited. The transactions data are especially useful for certain COICOP categories, part of which may normally be asked in online questionnaires and not in the diaries. For such expenditures, the data may give reduction of burden. Furthermore, the data may serve as a reminder of expenditures. In order to get consent, instructions and motivations should be very clear on what respondent effort is removed and/or how data are enriched. The app frontend is complicated by the multitude of banks and the need for bank account numbers. Also the bank transactions data need to be translated to products and COICOPS. Given that bank transactions data will only apply to a subset of the expenditures and a set of decision rules will have to be applied about completeness of data, the backend is complex.

Loyalty card data resemble scanner data. However, data is requested by the respondent and not the NSI, which means that the backend can easily be adapted. In @HBS test rounds, test persons were very positive about this option. We expect that willingness would be high. For stores that have loyalty card programs, the respondent effort would be greatly reduced. He/she only needs to provide a loyalty card number. The downside is, however, that there are many stores and respondents would have to provide numbers for all stores. They should also be able to remove consent. The frontend becomes more complicated due to the consultation of store API's/websites and the mix of stores without and with loyalty card programs.

All four big data solutions have their strong and weak features. Since barcode scanning still is burdensome and demands for a complex backend, this option was not considered within @HBS. The scanner data option is very dependent on the negotiations with a wide range of stores and also demands for a complex backend. This option was explored but not deemed feasible unless stores move to digital receipts. The other options, bank transactions data and loyalty card data, may be the most promising as they, essentially, lean on a "backend" at banks and stores.

5.3 Geo-locations

The third smart feature employs location sensors (GPS, Wi-Fi, GSM) in mobile devices. Time-location data can be measured and stored by devices after explicit respondent consent and may serve two purposes: 1) assist respondents by providing overviews of visited locations where expenditures are likely, and 2) provide information to link big data, in particular scanner data.

Location measurements have imperfections. Location measurements have an accuracy that depends on the type of sensor. GPS is the most accurate but can be blocked by buildings and constructions. Wi-Fi is the second most accurate, but is only available in or near buildings. GSM is the leas accurate but almost always available. A mobile device does not need to connect to internet or to Wi-Fi to detect GPS or Wi-Fi.
GPS location measurements cost device battery. For this reason, modern operating systems limit the access of apps to GPS when the app runs on the background. This can to some extent be avoided by disabling battery reduction in the mobile device settings. Nonetheless, missing data occurs.

For the first purpose, identification of expenditure locations, two approaches exist: geo-fencing and stop detection. Geo-fencing means that beforehand a database of coordinates of points-of-interest is prepared. Every time a respondent approaches such a coordinate within a pre-specified distance, the app is triggered and may list or sent a notification. Online databases of points-of-interest are available and contain some metadata about the locations. Stop detection means that location-time measurements are clustered to tracks and stops. A stop is detected when the mobile device stays within a specified perimeter for at least a specified amount of time. The perimeter and time parameters can be tuned, making stop detection more or less sensitive. The advantage of geo-fencing over stop detection is the robustness to temporary inaccuracy or missing data of location measurements. The advantage of stop detection over geo-fencing is the absence of need for a database of points-of-interest.
In HBS, both options can be implemented and three strategies to employ the data have been identified within @HBS:

1. Map in app UI: Time-location measurements are made on a constant basis and respondents can consult their travels throughout the day on a map provided on a separate tab of the app UI. They then have to decide themselves whether they visited locations or points-of-interest where they made an expenditure.
2. List in app UI: The app only lists stops on a separate tab of the app UI. Respondents can go through the stops as a reminder to where they have (as predicted by the location measurements).
3. Notification: Once a stop is detected or a geo-fence is passed, the app sends notifications to the respondent, asking her/him whether an expenditure was made.

The second purpose of time-location measurements, the link to scanner data, is strongly dependent on the access to individual transactions. This option was discussed in section 5.2 and not deemed realistic until digital receipts are introduced and/or stores have a backend ready for data requests. It was not further explored within @HBS.

The geo-location feature was scored on the same criteria as the four big data options in section 5.2,:respondent burden reduction, respondent willingness, frontend complexity and backend complexity. See table 5.2 for the subjective scores. Reduction in respondent burden is modest as it does not remove tasks but only reminds respondents of tasks. In general, there is some privacy concern about time-location measurements. This may be outweighed by the utility for the survey; for example, in an app-assisted travel survey at Stat Netherlands around 30% of the sample consented to time-location data and around 20% kept the app going for at least a week. In HBS the rationale for time-location data is weaker, which will affect willingness. The app frontend is relatively complex as any information on stops should be included in the app UI without distracting the main tasks. The app backend is relatively straightforward as the time-location data do not have to be stored and kept.

Table 5.2: Scores of geo-locations on reduction in respondent burden, willingness to consent, complexity of app frontend and complexity of app backend. Burden and willingness are rated on a three-point scale of + to +++ with +++ the largest reduction of burden. Frontend and backend complexity are rated on a three-point scale from – to -— with --- the most demanding in development and maintenance.

	Reduction in respondent burden	Respondent willingness to consent	Complexity of app frontend	Complexity of app backend
Geo-location	+	+	--	-

Based on the scores, it was decided within @HBS to prioritize receipt scanning and big data linkage over geo-locations. Given past experiences, it would be relatively straightforward how to include time-location in the frontend and backend of the HBS app. The most natural approach is geo-fencing, which is less sensitive to inaccurate and missing location measurements.

6. Future development

Based on the test rounds in three countries, we conclude that an app-assisted approach towards the HBS is a very promising and feasible route. We recommend that this approach is further elaborated and expanded. Out of the three smart features, receipt scanning, big data linkage and geo-locations, the first turned out very useful and the second sufficiently promising to refine and test. Receipt scanning and scanner data linkage have a close resemblance to on-going research by Eurostat into iCards/NFC-type payments where shop transactions are flagged and sent to the NSI. We recommend that in further development, these options are closely monitored in conjunction with each other.

From January 2020 to December 2021, ESSnet Smart Surveys will explore smart surveys and trusted smart surveys and provide specifications to collection platforms for such surveys. One of the four survey themes in the ESSnet is consumption and the HBS app will feature in usability tests and field tests across four to six countries. The ESSnet is not intended to further develop apps, but will, obviously, bring new results and experience that will help improve the app. In the ESSnet field test, two experimental conditions are investigated: interviewer assistance in recruiting, helping and motivating respondents to use an app, and the type of feedback given to the household on their expenditures.

As the main follow-up topics for development of the HBS app, we see the following:

- Receipt scanning pipeline: It is likely that the receipt scan processing needs a few iterations to be perfected. At first, it will be assumed that most of the processing needs to be done in-house, but gradually tasks may be performed in-app;
- Big data consent questions: This is a promising option to reduce respondent burden and one that should be explored further;
- Concepts/user-centered approach: App-assisted instruments will benefit strongly from a reconsideration of concepts and terminology within a user/respondent-centered approach.
- Multi-devices/modes: It is likely that multiple devices and modes will have to be combined in order to ensure good representation of the population;

As a stepping stone to such research and development, Stat Austria will conduct analyses on field data they have collected in 2019 with the Konsumerhebung app. The Household Budget Survey app has partly been modelled after this app.. Stat Austria will focus on three topics:

- Representation and devices: In the Austrian app, respondent scan choose the type of device, i.e. desktop/laptop, tablet and smartphone. Relevant questions are what devices households prefer and, if and how this preference is related to their characteristics.
- Protocol adherence: The various devices are synchronized, so that device switches are possible. After respondents have chosen a device, the relevant questions are whether they use different devices for different tasks and whether they switch devices during data collection. Subsequent questions are whether design features of the data collection strategy affect this behaviour.
- Data quality: Perhaps, most importantly, does device choice affect the richness and completeness of expenditure data.

Findings and results of project @HBS will be presented at various conferences and workshops. Future plans and development will be discussed at these occasions.

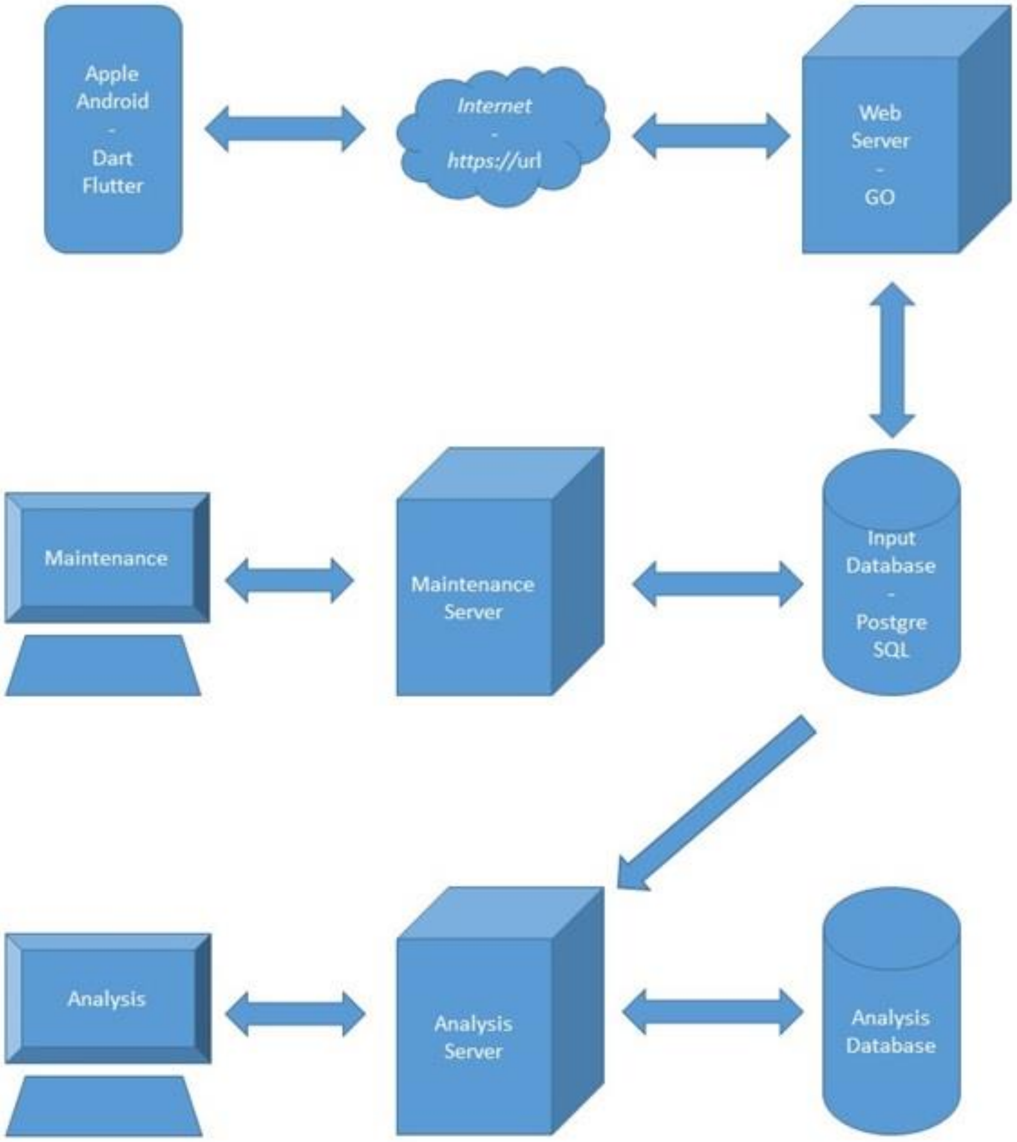
Appendix A – Overview of @HBS supplemental products

List of detailed products of @HBS:

1. Test material round 1, WP3, Giesen, D., Theunissen, S., Nyholt, M., Vrabič-Kek, B., Zgonec, M., Järvensivu, M., Niemelä, A.
2. Test plan, WP3, Giesen, D.
3. Test protocol round 2, WP3, Giesen, D., Theunissen, S., Nyholt, M., Vrabič-Kek, B., Zgonec, M., Järvensivu, M., Niemelä, A
4. Test report round 1 + 2, WP3, Vrabič-Kek, B., Giesen, D., Järvensivu, M.
5. Test mock-ups for big data linkage consent, WP3, Giesen, D., Theunissen, S.
6. Comparison of in-app and in-house receipt processing, WP4, Benedikt, L., Joshi, C., Nolan, L., Wolf, N. de, Schouten, B.
7. Installation guide, WP2, Theunissen, S.
8. Recruitment flyer HBS, WP3, Theunissen, S.
9. Invitation letter HBS, WP3, various authors at NL
10. Source code HBS app, WP2, Oerlemans, T., Janssen, J.
11. Backend specifications, WP2, Janssen, J.
12. CSPA documentation, WP2, Janssen, J., Oerlemans, T., Schouten, B.

Appendix B – Backend specifications

Household Budget Survey - Mobile App - System overview



System parts:

- Users enter/manage data on Apple or Android smartphones with HBS app. HBS app is developed in Dart/Flutter.
- Communication over internet via secure HTTPS URL.
- Web Server monitors HTTPS URL and synchronizes data on smartphones with data in input database. User can manage data on multiple smartphones. Data is timestamped on smartphone, data with latest timestamp is valid. Web server is developed in GO language.

- Input Database stores and synchronizes data from smartphones. Also used for management of usernames and passwords (hashed). Input database is developed in PostgreSQL.

System parts vision (to be implemented in Q2 2020):

- Maintenance and monitoring of user response, management of passwords.
- Maintenance Server for communication between maintenance and input database.
- Analysis of data by statisticians.
- Analysis Server for analysis of data in analysis database, and data import from input database to analysis database.
- Analysis Database for final data storage and data analysis.

Functionality

The main functionality of the HBS backend is:

- User authentication: check whether the user password matches the (hashed) password in the database
- Store receipt data: the data gathered on the smartphone is also stored in the database.
- Synchronize receipt data: keep the data synchronized between smartphones if the user uses multiple smartphones.
- Monitoring: respondent activity and app usage is monitored and may be input to follow-up actions

The HBS-App maintains the receipt data locally on a phone. When the HBS-App is started, or when a receipt is entered or changed the app tries to connect to the GO server to synchronize the locally stored data with the data in the database. The HBS-App does not need a continuous connection with the GO server, when there is a connection all data will be synchronized.

Status

The Dutch Statistics office is in a transition going towards an infrastructure that is delivered through Pivotal Cloud Foundry. We aimed at releasing the HBS backend with this new infrastructure. Unfortunately this infrastructure is not yet fully operational therefore we could not test a full working backend.

We did manage to test the backend on a Pivotal Cloud Foundry playground outside of the Dutch Statistics office.

- the database size on this playground was only 20 MB.
- the URL we used for this test was not secure (HTTP instead of HTTPS).

The backend was tested on a small scale (one user with two smartphones, max. 4 photos). On this small scale (using a HTTP URL) everything worked fine.

PostgreSQL Input Database

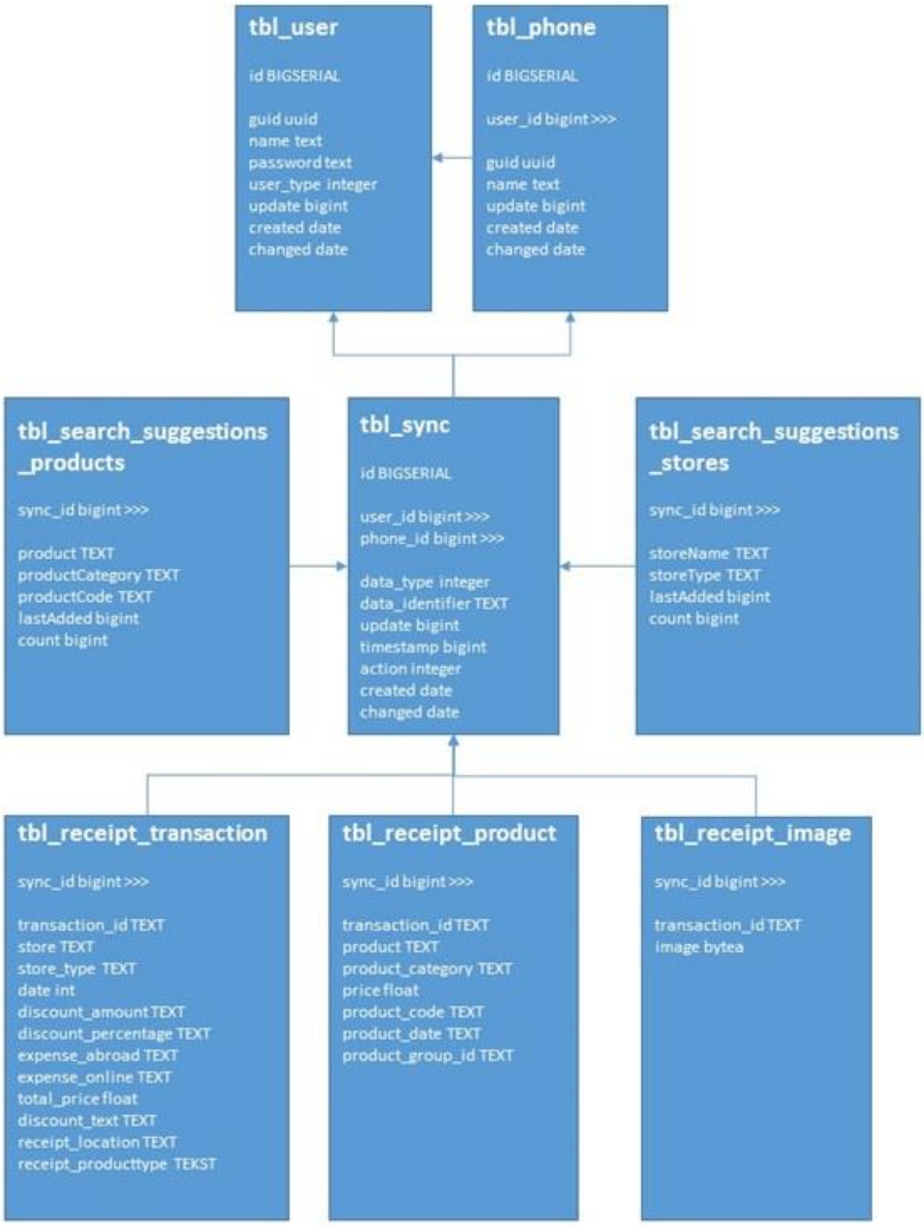
The PostgreSQL Input database contains eight tables (see data model below):

• tbl_user	: username + hashed password
• tbl_phone	: multiple phones per users
• tbl_sync	: data synchronization mechanism
• tbl_receipt_transaction	: receipt information
• tbl_receipt_product	: receipt products
• tbl_receipt_image	: receipt photos

- tbl_search_suggestions_products : frequently used products
- tbl_search_suggestions_stores : frequently used stores

There is no further functionality in the database. The contents of these tables is managed by the GO server. The creation script for these tables is located in the file structure of the GO-language server. It is possible to initialize the database (add users + hashed passwords) with the GO-language server.
The required size of the database depends on the number of users, the length of the survey period and the number of receipts a user enters per day. Per receipt the required database size is: +/- 5 MB, this is roughly the size of a photo + 1KB of textual data. Example: If 100 users take 2 photos per day for a period of one month, the required database size is: 100 * 2 * 5 MB * 30 = 30.000 MB = 30 GB.

Household Budget Survey Input data Model.



tbl_user
id BIGSERIAL
guid uuid
name text
password text
user_type integer
update bigint
created date
changed date

tbl_phone
id BIGSERIAL
user_id bigint>>>
guid uuid
name text
update bigint
created date
changed date

tbl_search_suggestions_products
sync_id bigint>>>
product TEXT
productCategory TEXT
productCode TEXT
lastAdded bigint
count bigint

tbl_sync
id BIGSERIAL
user_id bigint>>>
phone_id bigint>>>
data_type integer
data_identifier TEXT
update bigint
timestamp bigint
action integer
created date
changed date

tbl_search_suggestions_stores
sync_id bigint>>>
storeName TEXT
storeType TEXT
lastAdded bigint
count bigint

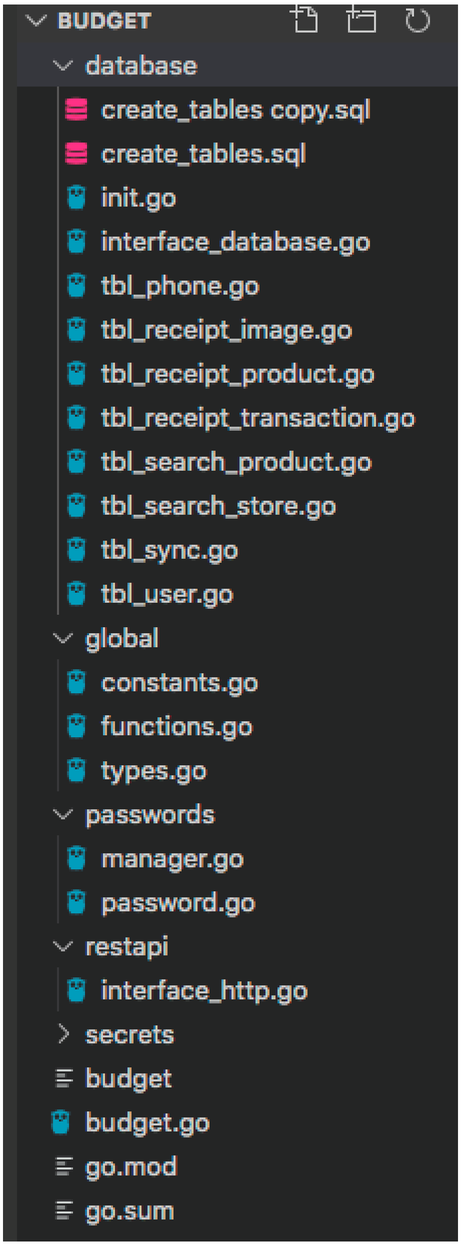
tbl_receipt_transaction
sync_id bigint>>>
transaction_id TEXT
store TEXT
store_type TEXT
date int
discount_amount TEXT
discount_percentage TEXT
expense_abroad TEXT
expense_online TEXT
total_price float
discount_text TEXT
receipt_location TEXT
receipt_producttype TEKST

tbl_receipt_product
sync_id bigint>>>
transaction_id TEXT
product TEXT
product_category TEXT
price float
product_code TEXT
product_date TEXT
product_group_id TEXT

tbl_receipt_image
sync_id bigint>>>
transaction_id TEXT
image bytea

GO-language server

File structure of GO server for the HBS-App:



```
BUDGET
  database
    create_tables copy.sql
    create_tables.sql
    init.go
    interface_database.go
    tbl_phone.go
    tbl_receipt_image.go
    tbl_receipt_product.go
    tbl_receipt_transaction.go
    tbl_search_product.go
    tbl_search_store.go
    tbl_sync.go
    tbl_user.go
  global
    constants.go
    functions.go
    types.go
  passwords
    manager.go
    password.go
  restapi
    interface_http.go
  secrets
  budget
  budget.go
  go.mod
  go.sum
```

- PostgreSQL script for creating tables
- Initial connection with database
- Main interface with database
- Standard functions per table (insert, update, delete, select etc.)

- General constants
- General functions
- General Types

- Create initial users + hashed passwords
- Password generation

- Rest-api URL Interface

- Secrets folder with users + passwords
- Compiled GO server executable
- Main entry of GO server
- Used modules
- Module summary

The main entry point for the GO server budget.go is initialized via environment settings:

- PORT : http port
- BUDGET_HOST : database host
- BUDGET_DBNAME : database name
- BUDGET_PORT: : database port
- BUDGET_USER : database user
- BUDGET_PASSWORD : database password

These environment settings are required and have to be set for your environment.

The GO server can be used to generate a set of standard usernames and passwords. See the readme section in passwords/manager.go. The generated usernames and hashed password are used to initialize the database. Also, in the folder secrets, files will be generated with usernames and passwords that can be communicated with users.

Future steps (Q2 and Q3 2020)

The HBS backend as it is, is only equipped to authenticate users and store receipt data in an input database. Additional functionality is not yet available.

The not yet implemented part of the system overview is:

- Maintenance and monitoring of user response, management of passwords.
- Maintenance server for communication between maintenance and input database.
- Data analysis by substantive researchers and methodology
- Analysis server for analysis of data in analysis database, and data import from input database in analysis database.
- Analysis database for final data storage and data analysis.

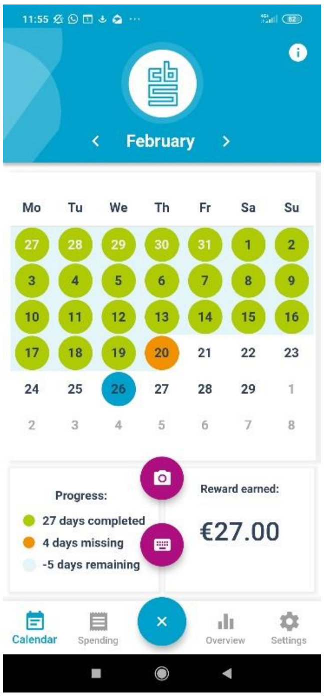
While this functionality is not available the input database can be managed/queried directly with the pgAdmin_4 tool for PostgreSQL databases.

Additional functionality:

- Analyze photos: create fully classified receipts from photos of receipts. This functionality can be added to the web server or (if required) another server with access to the input database.
- App usage data: at this moment we only gather data of receipts. We are interested in:
    - Type of phone used
    - Dwell time on app pages
    - Technical problems/crashes
    - Use of certain help options
    - Consultation of personal statistics in the app

Appendix C – Screenshots HBS app

Calendar:



Mo	Tu	We	Th	Fr	Sa	Su
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	1
2	3	4	5	6	7	8

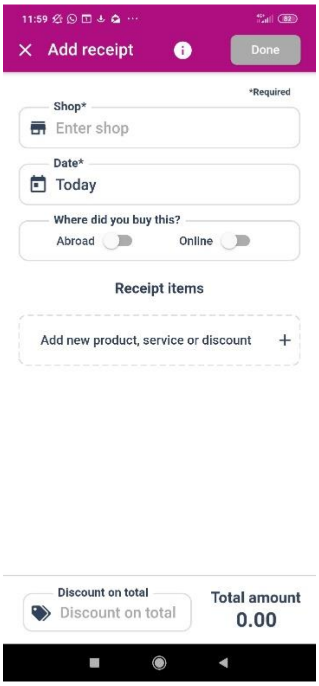
Progress:

- 27 days completed
- 4 days missing
- -5 days remaining

Reward earned:

€27.00

Manual data entry:



Shop*: Enter shop

Date*: Today

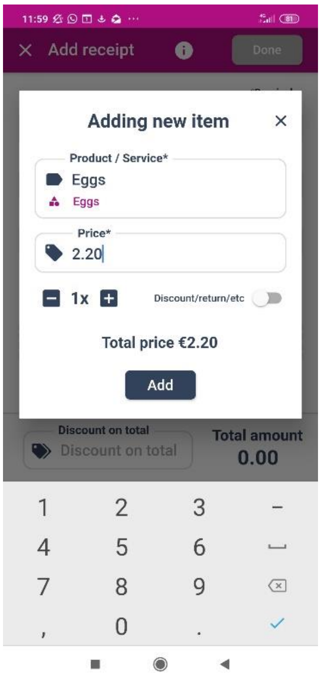
Where did you buy this? Abroad Online

Receipt items

Add new product, service or discount +

Discount on total: Discount on total

Total amount 0.00



Adding new item

Product / Service*: Eggs
Eggs

Price*: 2.20

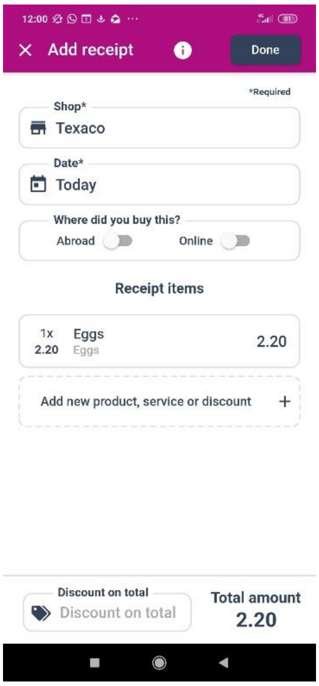
1x Discount/return/etc

Total price €2.20

Add

Discount on total: Discount on total

Total amount 0.00



Shop*: Texaco

Date*: Today

Where did you buy this? Abroad Online

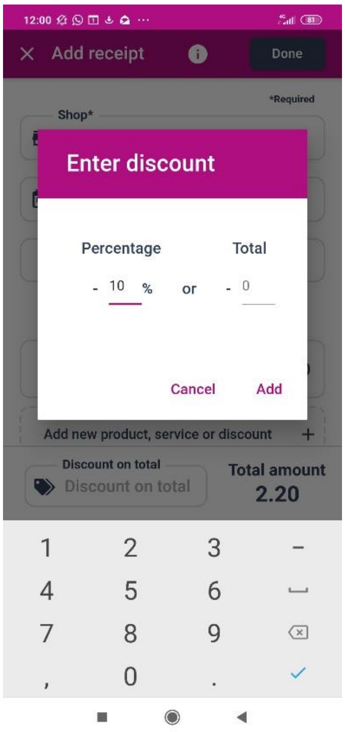
Receipt items

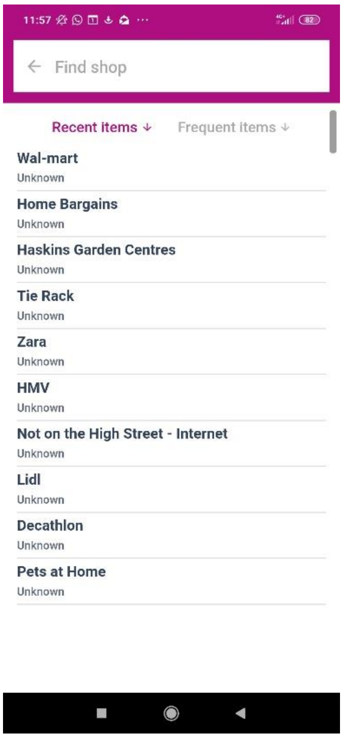
1x 2.20 Eggs Eggs 2.20

Add new product, service or discount +

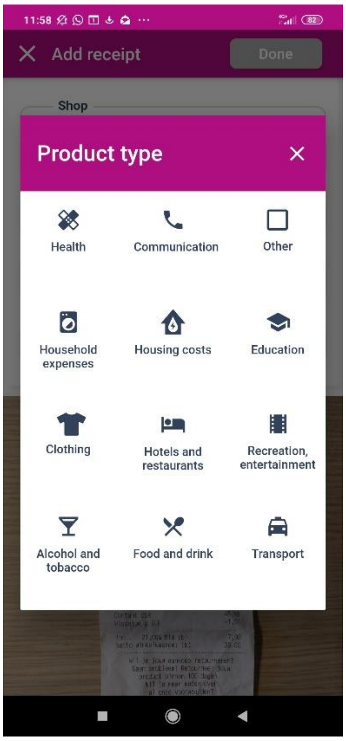
Discount on total: Discount on total

Total amount 2.20

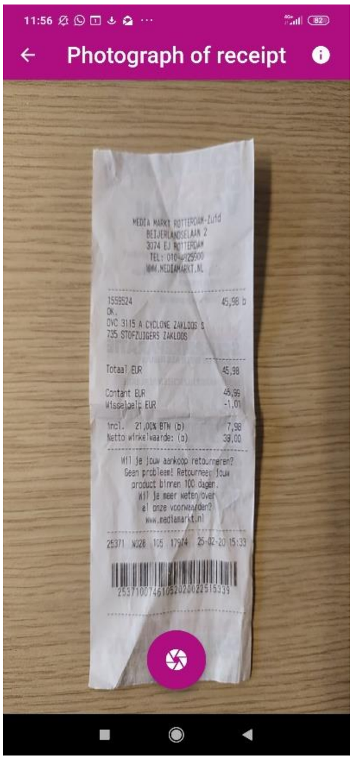


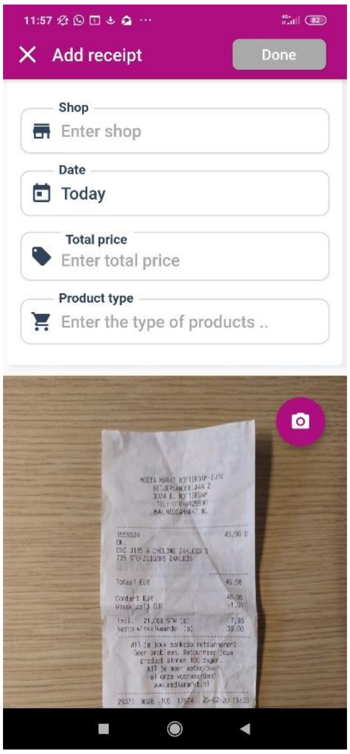


Wal-mart
Unknown
Home Bargains
Unknown
Haskins Garden Centres
Unknown
Tie Rack
Unknown
Zara
Unknown
HMV
Unknown
Not on the High Street - Internet
Unknown
Lidl
Unknown
Decathlon
Unknown
Pets at Home
Unknown

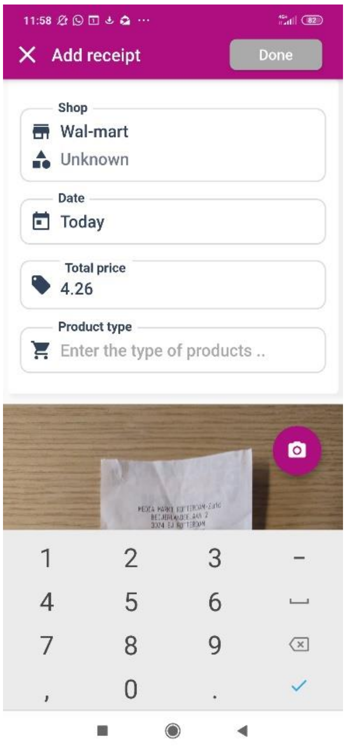


Scan data entry:



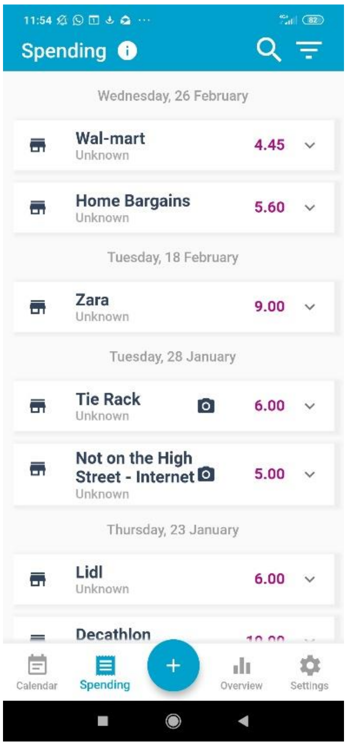


Shop
Enter shop
Date
Today
Total price
Enter total price
Product type
Enter the type of products ..



Shop
Wal-mart
Unknown
Date
Today
Total price
4.26
Product type
Enter the type of products ..

Expenditure overview:



11:54

Spending

Wednesday, 26 February

Wal-mart — Unknown — 4.45

Home Bargains — Unknown — 5.60

Tuesday, 18 February

Zara — Unknown — 9.00

Tuesday, 28 January

Tie Rack — Unknown — 6.00

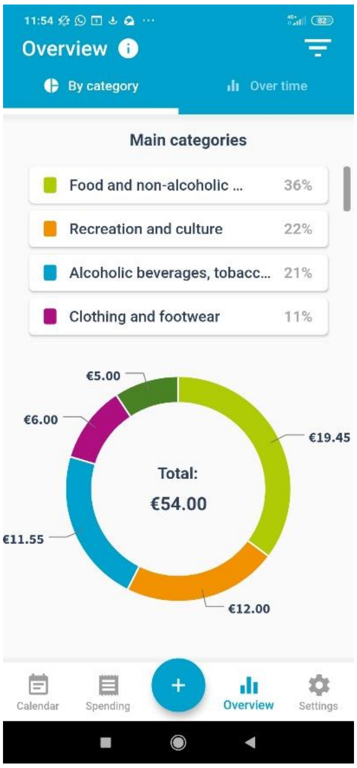
Not on the High Street - Internet — Unknown — 5.00

Thursday, 23 January

Lidl — Unknown — 6.00

Decathlon — 10.00

Statistics overview:



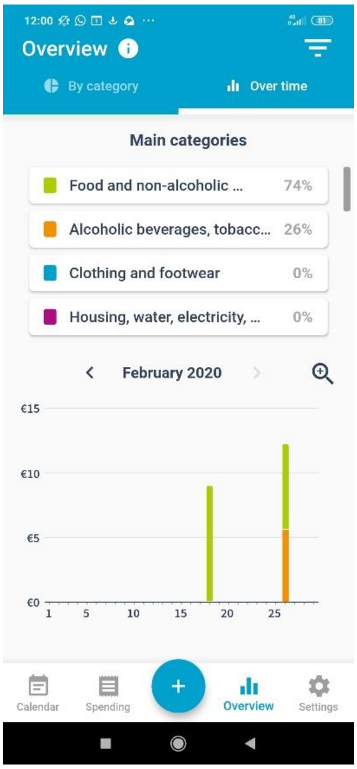
11:54

Overview

By category | Over time

Main categories

🟩	Food and non-alcoholic ...	36%
🟧	Recreation and culture	22%
🟦	Alcoholic beverages, tobacc...	21%
🟪	Clothing and footwear	11%



12:00

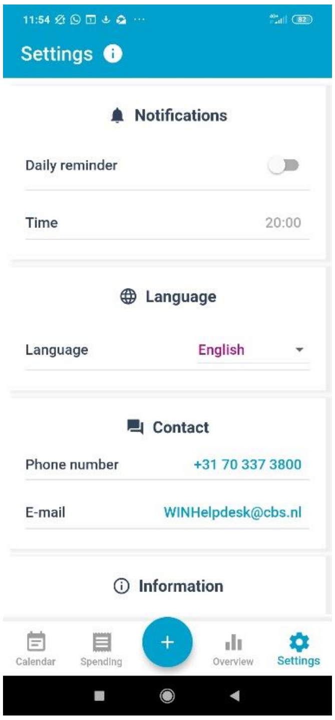
Overview

By category | Over time

Main categories

🟩	Food and non-alcoholic ...	74%
🟧	Alcoholic beverages, tobacc...	26%
🟦	Clothing and footwear	0%
🟪	Housing, water, electricity, ...	0%

Settings:



Settings

Notifications

Daily reminder

Time 20:00

Language

Language English

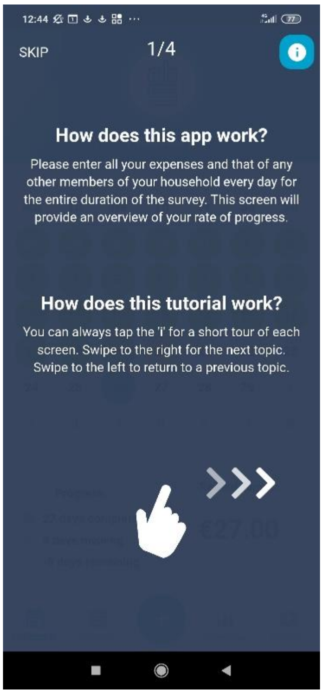
Contact

Phone number +31 70 337 3800

E-mail WINHelpdesk@cbs.nl

Information

Tutorial:



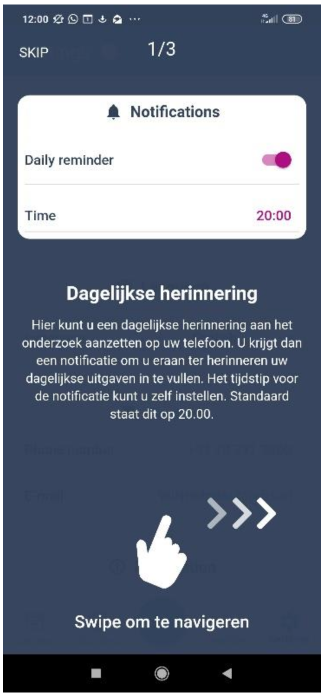
SKIP 1/4

How does this app work?

Please enter all your expenses and that of any other members of your household every day for the entire duration of the survey. This screen will provide an overview of your rate of progress.

How does this tutorial work?

You can always tap the 'i' for a short tour of each screen. Swipe to the right for the next topic. Swipe to the left to return to a previous topic.



SKIP 1/3

Notifications

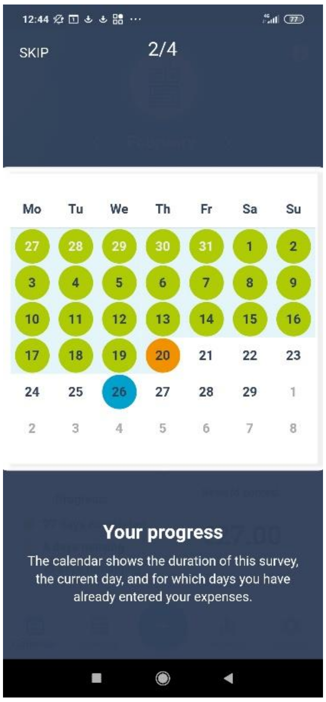
Daily reminder

Time 20:00

Dagelijkse herinnering

Hier kunt u een dagelijkse herinnering aan het onderzoek aanzetten op uw telefoon. U krijgt dan een notificatie om u eraan ter herinneren uw dagelijkse uitgaven in te vullen. Het tijdstip voor de notificatie kunt u zelf instellen. Standaard staat dit op 20.00.

Swipe om te navigeren



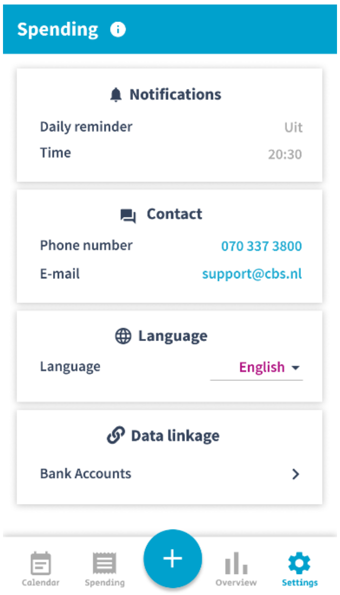
SKIP 2/4

Mo	Tu	We	Th	Fr	Sa	Su
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	1
2	3	4	5	6	7	8

Your progress

The calendar shows the duration of this survey, the current day, and for which days you have already entered your expenses.

Appendix D – Screenshots HBS app bank mockups


Spending

Notifications

Daily reminder Uit

Time 20:30

Contact

Phone number 070 337 3800

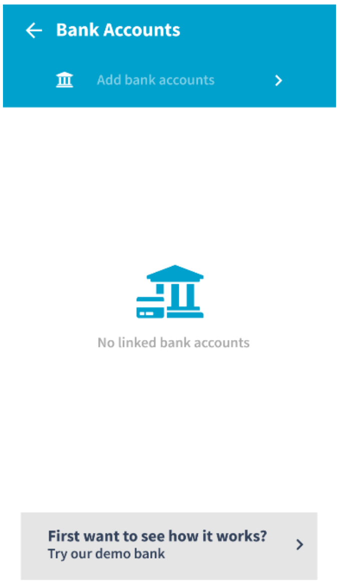
E-mail support@cbs.nl

Language

Language English

Data linkage

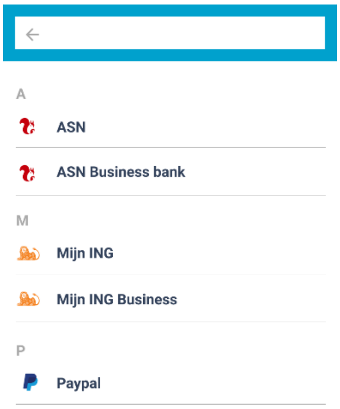
Bank Accounts


Bank Accounts

Add bank accounts

No linked bank accounts

First want to see how it works?
Try our demo bank


A

ASN

ASN Business bank

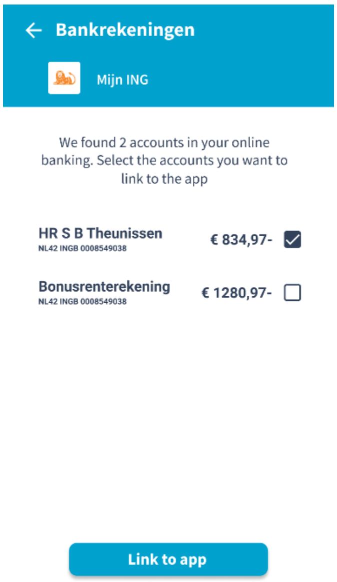
M

Mijn ING

Mijn ING Business

P

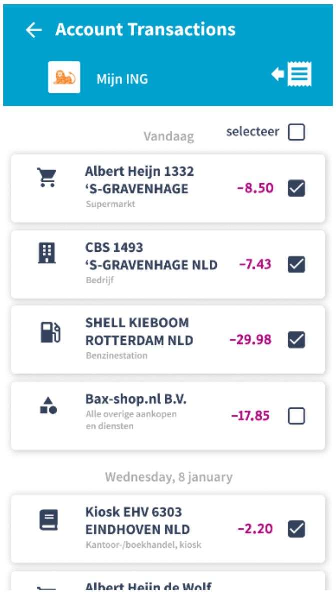
Paypal


Bankrekeningen

Mijn ING

We found 2 accounts in your online banking. Select the accounts you want to link to the app

HR S B Theunissen NL42 INGB 0008549038	€ 834,97-	☑
Bonusrenterekening NL42 INGB 0008549038	€ 1280,97-	□


Account Transactions

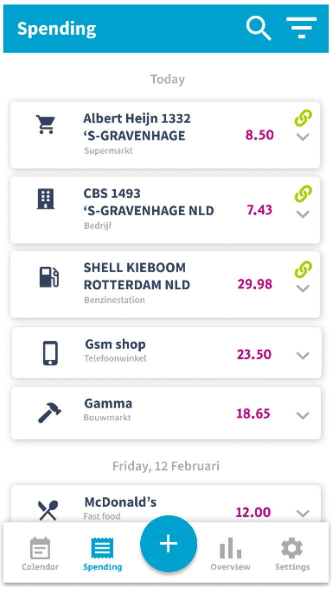
Mijn ING

Vandaag selecteer □

Albert Heijn 1332 ‘S-GRAVENHAGE Supermarkt	−8.50	☑
CBS 1493 ‘S-GRAVENHAGE NLD Bedrijf	−7.43	☑
SHELL KIEBOOM ROTTERDAM NLD Benzinestation	−29.98	☑
Bax-shop.nl B.V. Alle overige aankopen en diensten	−17.85	□

Wednesday, 8 january

Kiosk EHV 6303 EINDHOVEN NLD Kantoor-/boekhandel, kiosk	−2.20	☑
Albert Heijn de Wolf		


Spending

Today

Albert Heijn 1332 ‘S-GRAVENHAGE Supermarkt	8.50
CBS 1493 ‘S-GRAVENHAGE NLD Bedrijf	7.43
SHELL KIEBOOM ROTTERDAM NLD Benzinestation	29.98
Gsm shop Telefoonwinkel	23.50
Gamma Bouwmarkt	18.65

Friday, 12 Februari

McDonald’s Fast food	12.00