# Requirements

## Contents


**1 Selection of Elicitation Technique** ..... 2

1.1 Interview ..... 2

1.2 Observation ..... 3

1.3 Brainstorming ..... 3

1.4 Survey/Questionnaire ..... 4

1.5 Choice of Elicitation Technique ..... 4

**2 Final Prioritized Set of Requirements** ..... 5

2.1 Dashboard App Requirements ..... 5

2.2 Account System Requirements ..... 6

2.3 Machine Learning App Requirements ..... 7

2.4 General Requirements ..... 7

**3 Additional Useful Artefacts used during Requirement Elicitation** ..... 9

3.1 Context Diagrams ..... 9

3.2 Use Case Diagram ..... 10

3.3 Detailed Use Cases ..... 10

**4 References** ..... 13

# 1 Selection of Elicitation Technique

In order for the project requirements to be effectively gathered, an appropriate elicitation technique is required to do this. The Business Analysis Body of Knowledge (BABOK)[1] highlights various different techniques, the four most common of which include interviews, observation, brainstorming, and the use of a survey/questionnaire. Further details of each technique can be found below:

## 1.1 Interview

Interviews allow for the elicitation of information via a conversation with a person or group of people. The interviewee, usually a stakeholder, would be asked a series of questions and their responses would be documented. These responses would help formulate the requirements needed for the application. There are two types of interviews that can be used: structured or non-structured[2]. The former having a pre-defined set of questions while the latter favouring an open discussion. A mix of these two types can also be implemented depending on the situation. This elicitation technique is useful as it allows the development team to find out exactly what a stakeholder would need, thus helping in the formation of the requirements.

The process of using interviews consists of three elements[3]:

1. **Preparation**: In this element the interview focus would be defined, and the interview structure would be chosen. This consists of forming the questions, whether open or closed, organizing the order of questions, and selecting the location and time of the interview.
2. **Conducting**: In this element the interview would be undertaken and the goals and questions previously decided should be met and discussed. The results would then be recorded or scribed.
3. **Follow-up interview**: In this element all information would be gathered and processed. The requirements would then be formed. The interviewee would also be sent a review and a thank you. If the requirements formed are not specific enough however, the interviewee could be asked back to provide some additional information.

*Table 1 – Advantages and Disadvantages of Interviewing*

| Advantages | Disadvantages |
|---|---|
| • Allows interviewer to establish a relationship with the interviewee (stakeholder)<br>• The interview is conducted in private, allowing the interviewee to answer questions more openly<br>• The interviewer can observe body language and tone to understand more from the answers than just the words. Helping to give more insight than other techniques which do not use face to face contact[4]. | • All three elements of the interview process can be time consuming.<br>• The interviewer would need to be trained which can be expensive and time consuming.<br>• Multiple interviews would need to be conducted for the different stakeholders which is again expensive and time consuming.<br>• The answers given in an open discussion type interview are more subject to interpretation[5]. |

## 1.2 Observation

Observation employs the use of a business analyst. The analyst assesses how a user does their job by shadowing their work environment and daily activities. This technique helps to provide context to requirements. Observation can involve watching over someone doing typical activities, documenting, and asking questions for explanation on completion of tasks[6].

*Table 2 - Advantages and Disadvantages of Observation*

| Advantages | Disadvantages |
|---|---|
| • Effective form of elicitation, allowing to assimilate more information quickly and confirm understanding as you go.<br>• As you ask follow-up questions and shadow varying activities, you might understand the full process without doing any work[7].<br>• Mainly important in providing context to requirement. | • The work proceedings may not encompass the whole of the project requirements.<br>• It restricts what you learn to what is already implemented, lacking creativity.<br>• It requires processes already in place and does not support start ups or new technologies. |

## 1.3 Brainstorming

This is where the group comes together to facilitate creative thinking in order to come up with new ideas based upon the problem description. There is no limit to the number of requirements and usually, teams try to come up with as many as possible which can then be later sorted in the prioritization stage. This technique allows for a strong focus on what the group know they are able to do in order to solve the problem, but since it does not involve the client or user, there may be some important requirements that get missed by the development team[8].

In general, brainstorming can be separated into three main sections: Preparation, Session, and Wrap-up[9]:

1. **Preparation**: The area of focus needs to be defined here so that the ideas are relevant to the problem. The criteria for judging each idea is also established so that each member knows what constitutes a useful idea.
2. **Session**: This is where the group shares the ideas together in the brainstorming session. The ideas should be recorded visually for all members as certain ideas can facilitate new ideas from other members. There should be no limit on the number of ideas to try and get as many useful ones as possible.
3. **Wrap-up**: The ideas are discussed individually. Duplicate ideas are removed and similar ideas can be combined with others. The ideas can then be ordered in terms of relevance and priority.

*Table 3 - Advantages and Disadvantages of Brainstorming*

| Advantages | Disadvantages |
|---|---|
| • Many ideas can be elicited in a short period of time.<br>• Creative thinking allows for novel approaches to situations[10].<br>• Since no external sources are required, the process can be quick, easy and cost effective. | • Depends heavily on the teams creativity.<br>• Since only members of the team participate, they may miss certain ideas which would be important to a user.<br>• Can be limited by the personalities and opinions of the group members[11]. |

## 1.4 Survey/Questionnaire

The survey or questionnaire technique consists of a set of questions that are given to stakeholders, based on existing systems or proposed requirements. Then the responses are collected, tabulated, and analysed to identify the interests of the involved stakeholders[12]. The questions can be closed ended, open ended and semi-closed ended:

1. **Closed ended:** These questions contain a predefined set of answers from which the respondent has to choose. Variations of the questions can be multiple choice, binary, ranking (from not important to very important), checklist response, production, number scaled and so on.
2. **Open-ended:** For these questions, freedom is given to the respondent to provide answers in their own words, which might be numeric or text. This technique is useful however interpreting the responses is time-consuming.
3. **Semi closed-ended:** These questions provide both the structure of the closed-ended questions and the flexibility of the open-ended questions, by adding free text boxes, such as when giving the "Other" option for multiple choice questions and a free text box may be displayed to provide additional details[13].

*Table 4 - Advantages and Disadvantages of Surveys/Questionnaires*

| Advantages | Disadvantages |
|---|---|
| • More accurate than an interview.<br>• Easy to get data from a large audience.<br>• Can be completed relatively quickly and accurately making them time effective.<br>• They can be versatile and relatively inexpensive. | • Questions might not be clear to all participants.<br>• Open-ended questions are time-consuming to interpret (hence more analysis required)[14].<br>• All of the stakeholders may not participate leading to biases in requirements.<br>• Follow-up questionnaires might be required (based on the answers of the first).<br>• Easy to introduce bias by poor survey design, poor survey administration and small response rates[15]. |

## 1.5 Choice of Elicitation Technique

Due to the limitations of the project, the chosen elicitation techniques are limited to those which do not involve external sources and users. This is because they require an ethics approval which is not possible to gain within the timeframe of the project. Noting this, it is clear that interviews and surveys cannot be used and hence the techniques are limited to **brainstorming** and **observation**. Therefore, both of these techniques were used to give a good representation for the project requirements. Brainstorming was initially used to gather the bulk of the requirements which was heavily tailored to what the team already knew about the project and what was deemed possible to implement. Once a sufficient amount was obtained, observation was employed to take inspiration from other similar dash and machine learning apps available online. This highlighted any useful features that were not apparent to the team previously but would also make a good addition to the project.

# 2 Final Prioritized Set of Requirements

The project requirements have been separated into various different groups based on their overall category within the project. This includes requirements related to the dashboard app, the machine learning app, the account system, an finally general requirements. Each requirement was specified using natural language[16] in the format:

`<id> The <system> shall <function>`

where `<id>` is the unique requirement identifier, `<system>` is the area of the project to which the requirement relates, and `<function>` is the task to be performed by the system. Each requirement was then categorized based on whether they are functional or non-functional. Functional requirements specify what the system does whereas non-functional requirements specify how the system should operate.

Finally the requirements were prioritized based on the MoSCoW prioritization technique[17]. This separates each requirement into four different categories: Must have – requirements which are necessary for the project to run, Should have – requirements which are not as vital but should still be completed, Could have – requirements which are not critical to the projects success but would make it better if there is any time at the end, and Won't have for now – requirements which are not worth implementing currently due to time and budget constraints but can be considered in the future.

## 2.1 Dashboard App Requirements

| Code | Requirement | Type | MoSCoW |
|---|---|---|---|
| RQ1 | The dash app shall provide historical cycle hire data in the form of a scatter plot. | Functional | M |
| RQ2 | The dash app shall provide simple and interactive graphs to visually show pricing data at any given time (to promote cycle usage). | Functional | M |
| RQ3 | The dash app shall take an input from the user asking for a postcode. And return the cycle hire data in and around that location | Functional | M |
| RQ4 | The dash app shall take an input from the user asking for a postcode. And return the pollution data in and around that location. | Functional | M |
| RQ5 | The dash app shall take an input from the user for a specific time period (past) and location and provide past cycle higher data. | Functional | M |
| RQ6 | The dash app shall be able to provide pricing information based on location and time (hour, day, season). | Functional | M |
| RQ7 | The dash app shall provide an explanation of each plot with its relevance in regard to the problem statement. | Functional | M |
| RQ8 | The dash app shall allow TFL to view how pricing is determined via the data used (clear and transparent). | Functional | S |
| RQ9 | The dash app shall display a map of London Boroughs with corresponding pollution levels. | Functional | S |
| RQ10 | The dash app shall take an input from the user of two locations and plan a route between them while avoiding areas with high pollution. | Functional | W |

| | | | |
|---|---|---|---|
| RQ11 | The dash app shall ask for users location and display it on the map if permission is given from the user. | Functional | W |
| RQ12 | The dash app shall indicate pollution levels (with written explanations) of their current location if permission is given from the user. | Functional | W |
| RQ13 | The dash app shall send notifications, if permission is given from the user, about calculated pricing based on time and pollution data. | Functional | W |
| RQ14 | The dash app shall ask for user location data to recommend cheaper routes based on current calculated pricing (dependant on time and area). | Functional | W |
| RQ15 | The dash app shall ask for permission to access user usage data to find the most popular routes and present further pricing development opportunities. | Functional | W |
| RQ16 | The dash app shall be able to receive inputs for newly recorded data from TFL and LAEI data collection services. | Functional | M |

## 2.2 Account System Requirements

| Code | Requirement | Type | MoSCoW |
|---|---|---|---|
| RQ17 | The dash app shall protect all customer data shared (Data Protection Act 2018)[18] | Non-Functional | M |
| RQ18 | The ML app shall allow users to create an account before they can access the app | Non-Functional | M |
| RQ19 | The dash app shall allow users to create an account before they can access the app | Non-Functional | M |
| RQ20 | The ML app shall check if the password entered by the user is strong enough by ensuring there is at least one capital letter and symbol and the length is over 8 characters | Non-Functional | S |
| RQ21 | The dash app shall check if the password entered by the user is strong enough by ensuring there is at least one capital letter and symbol and the length is over 8 characters | Non-Functional | S |
| RQ22 | The ML app shall allow users to login to their account in order to access the app | Non-Functional | M |
| RQ23 | The dash app shall allow users to login to their account in order to access the app | Non-Functional | M |
| RQ24 | The ML app shall send a confirmation email if a new account is created | Non-Functional | W |
| RQ25 | The dash app shall send a confirmation email if a new account is created | Non-Functional | W |
| RQ26 | The ML app shall allow users to log in with their Google accounts[19] | Non-Functional | W |
| RQ27 | The dash app shall allow users to log in with their Google accounts | Non-Functional | W |
| RQ28 | The dash app shall allow people from TFL and LAEI data collection services to create a different account with the permission to upload newly recorded data. | Non-Functional | C |

| | | | |
|---|---|---|---|
| RQ29 | The ML app shall allow people from LAEI data collection services to create a different account with the permission to upload newly recorded data. | Non-Functional | C |
| RQ30 | The ML app shall assign the user permissions based on their roles in the company | Non-Functional | C |
| RQ31 | The dash app shall assign the user permissions based on their roles in the company | Non-Functional | C |
| RQ32 | The ML app shall assign one time use forgot password tokens upon a user requesting a token after forgetting their password | Non-Functional | W |
| RQ33 | The dash app shall assign one time use forgot password tokens upon a user requesting a token after forgetting their password | Non-Functional | W |

## 2.3 Machine Learning App Requirements

| Code | Requirement | Type | MoSCoW |
|---|---|---|---|
| RQ34 | The ML app shall employ a simple machine learning algorithm to make a prediction of the pollution level given the location and date inputted by the user | Functional | M |
| RQ35 | The ML app shall perform without failure in 95 percent of use cases[20] | Non-Functional | S |
| RQ36 | The ML app shall restore in case of failure in no more than 15 minutes | Non-Functional | S |
| RQ37 | The ML app shall allow users to provide feedback through the app | Non-Functional | C |
| RQ38 | The ML app shall notify the user about any conflict with the requested location and date | Functional | S |
| RQ39 | The ML app shall allow the user to update the inputted location and date to resolve the conflict | Functional | S |
| RQ40 | The ML app shall be available in all hours of the day | Non-Functional | S |
| RQ41 | The ML app shall operate in a fair and unbiased manner[21] | Non-Functional | M |
| RQ42 | The ML app shall be easy to modify to improve performance or adapt to changes | Functional | C |
| RQ43 | The ML app shall repeatedly run the same algorithm and obtain the same or similar results | Functional | M |
| RQ44 | The ML app shall provide users with easy-to-read data | Functional | M |
| RQ45 | The ML app shall update the existing data when new data is added and use that when making a new prediction. | Functional | M |

## 2.4 General Requirements

| Code | Requirement | Type | MoSCoW |
|---|---|---|---|
| RQ46 | The dash app shall support computer devices running windows, Linux and macOS. | Non-Functional | M |
| RQ47 | The dash app shall meet the Web Content Accessibility Guidelines[22][23]. | Non-Functional | M |

| | | | |
|---|---|---|---|
| RQ48 | The dash app shall respond to users action under 3 seconds or less, including the rendering of text and images | Non-Functional | S |
| RQ49 | The dash app shall support both mobile phone and tablets. | Non-Functional | C |
| RQ50 | The dash app shall support iPhone devices running OS versions 3.4, 3.5 and 3.6. | Non-Functional | C |
| RQ51 | The dash app shall be scalable enough to support a large number visits at the same time | Non-Functional | W |
| RQ52 | The dash app shall update the existing data when a large number of new data is added in less than 10 minutes. | Non-Functional | S |
| RQ53 | The ML app shall support computer devices running windows, Linux and macOS. | Non-Functional | M |
| RQ54 | The ML app shall meet the Web Content Accessibility Guidelines. | Non-Functional | M |
| RQ55 | The ML app shall respond to users action under 3 seconds or less, including the rendering of text and images | Non-Functional | S |
| RQ56 | The ML app shall support both mobile phone and tablets. | Non-Functional | C |
| RQ57 | The ML app shall support iPhone devices running OS versions 3.4, 3.5 and 3.6. | Non-Functional | C |
| RQ58 | The ML app shall be scalable enough to support a large number visits at the same time | Non-Functional | W |
| RQ59 | The ML app shall update the existing data when a large number of new data is added in less than 10 minutes. | Non-Functional | S |

# 3 Additional Useful Artefacts used during Requirement Elicitation

In order to facilitate the elicitation of the requirements, several other diagrams and techniques were used to help outline the project. These included context diagrams, use case diagrams, and detailed use cases. Further details and explanations of each technique can be found below:

## 3.1 Context Diagrams

Context diagrams help set the boundary and scope of the project. It clearly shows the boundaries of the system as well as the flows of information (arrows) between different external entities (rounded rectangles) and the system itself (circle) which allows some direction for the requirements to be elicited.

Below in Figure 1 shows the context diagram for the machine learning app. This helped to highlight the idea of the London Atmospheric Emission Inventory (LAEI) being able to input new data directly into the system in order to increase the accuracy of any predictions made within the app. This helps to keep the data up to date and was not clear before the diagram was made. It also shows how the TfL staff interact with the system itself, highlighting the fact they will need to login with their email and password, as well as give a location and date input parameter to the machine learning model which provides them a prediction of the pollution level at the given date and location.

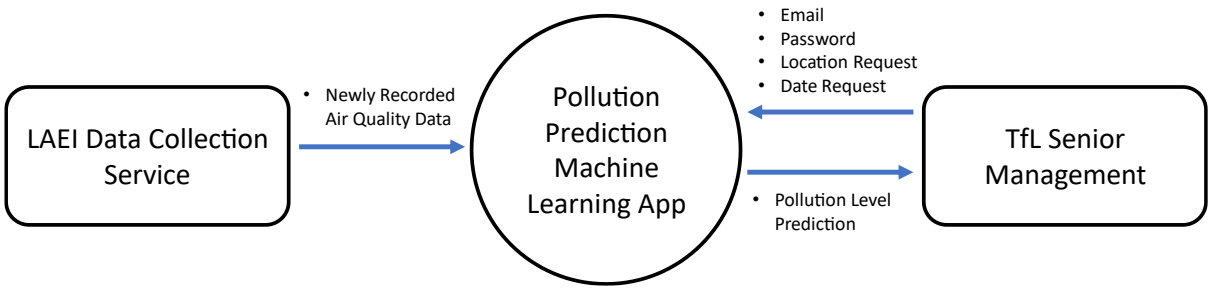


*Figure 1 – Context Diagram for the Machine Learning App*

In the same way, a context diagram was created for the dashboard app with the only difference being that TfL can also input newly recorded data, and the system provides air quality and cycle hire plots.

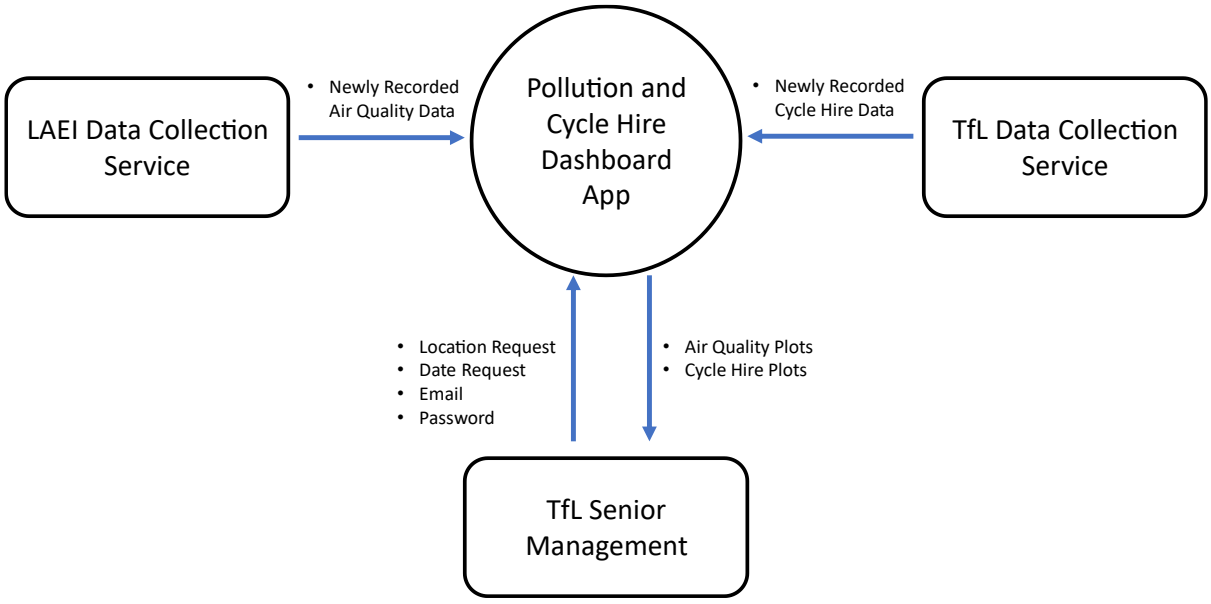


*Figure 2 – Context Diagram for the Dashboard App*

## 3.2 Use Case Diagram

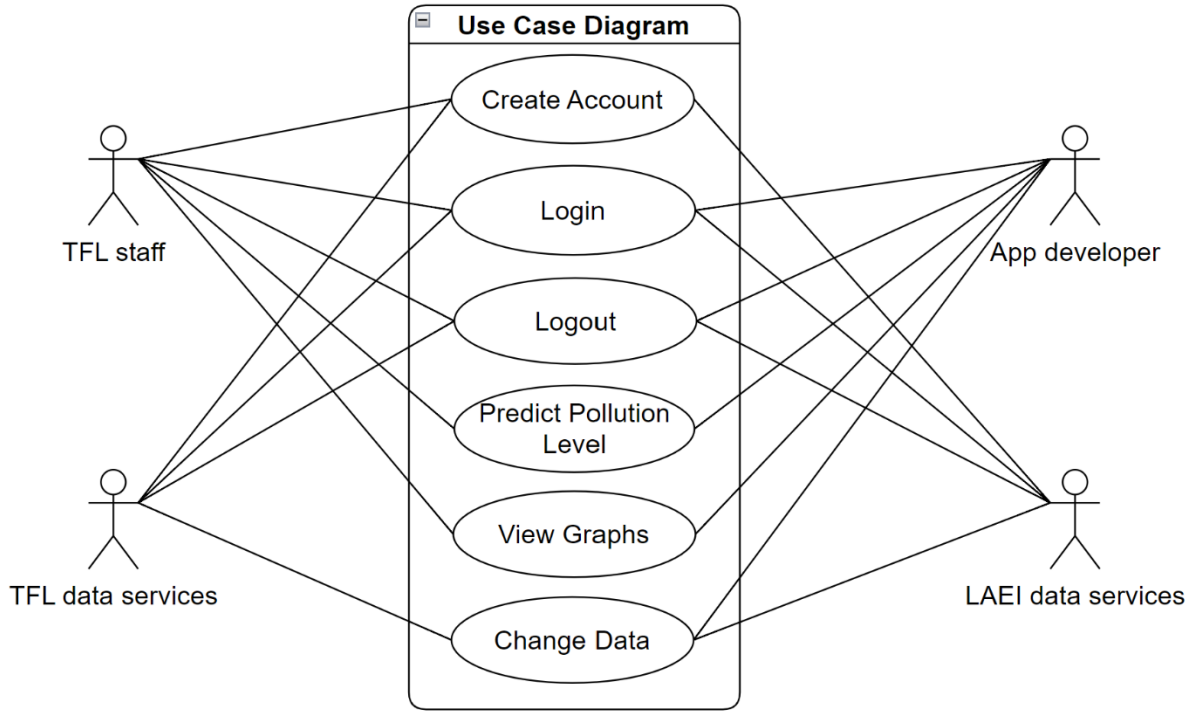


*Figure 3 – Use Case Diagram*

The UML Use Case Diagram seen above in Figure 3 focus on what the actors (stick figures) what to do in the apps. The actors are the TFL staff from the persona example, the app developers who would be editing and creating the app, and finally the TFL and LAEI data services who would be updating the existing data within the app. Focusing on the users themselves helps to focus the direction of the requirements. The diagram can be used in conjunction with detailed use cases found below.

## 3.3 Detailed Use Cases

Detailed Use Cases are made for each app function seen in the use case diagram. They give a brief description of each use case, explicitly state what actors are involved in the use case, the conditions pre-required in order for the use case to be undertaken, and finally the main and alternative flows in which the actor will interact with the use case. These can all be found below which again, helped give some more areas of focus for the requirements:

| **Use case 01** | **Login** |
|---|---|
| Brief Description | User logs in |
| Primary Actors | TFL staff, App developers, TFL data services, LAEI data services |
| Pre-conditions | User has an account |
| Main flow | 1. User enters email and password for log in<br>2. System checks if email and password matches to a registered account<br>3. If there is a match, user is logged in and sent to the Price Map page |
| Alternative flow | a. If only the email matches to a registered account, the user is informed that there is an error in the password<br>b. If the email does not match, the user is sent to the sign up page. |

| **Use case 02** | **Logout** |
|---|---|
| Brief Description | User logs out |
| Primary Actors | TFL staff, App developers, TFL data services, LAEI data services |
| Pre-conditions | User is logged in |
| Main flow | 1. User clicks log out<br>2. They are logged out and sent to the home page |
| Alternative flow | N/A |

| **Use case 03** | **Sign up** |
|---|---|
| Brief Description | User creates an account |
| Primary Actors | TFL staff, TFL data services, LAEI data services |
| Pre-conditions | N/A |
| Main flow | 1. User is prompted to fill in the Sign Up form<br>2. System checks if email given is a valid email.<br>3. If email is valid, account is created and user is redirected to the Price Map page |
| Alternative flow | a. If email is invalid, user is prompted with an invalid email message and requested to enter a different email. |

| **Use case 04** | **View graphs** |
|---|---|
| Brief Description | User accesses graph pages |
| Primary Actors | TFL staff, App Developers |
| Pre-conditions | User is logged in, user has access to page |
| Main flow | 1. User clicks on either the pollution graph or usage graph page<br>2. System checks if user has the appropriate permissions to access the clicked page<br>3. if user has permission, the user is sent to the chosen page |
| Alternative flow | a. If user does not have permission, the user is prompted with an invalid permission message |

| **Use case 05** | **Change data** |
|---|---|
| Brief Description | data is added or updated in the App |
| Primary Actors | TFL data services, LAEI data services, App developers |
| Pre-conditions | User is logged in, user can change data |
| Main flow | 1. User changes either pollution or usage data to the app<br>2. System checks if user has appropriate permissions and if data is in the correct form<br>3. If data is appropriate, system updates website pages with new data |
| Alternative flow | a. If user does not have permission, the user is prompted with an invalid permissions message<br>b. If Data is not appropriate, user is prompted with an invalid data prompt |

| **Use case 06** | **Predict pollution level** |
|---|---|
| Brief Description | System predicts the pollution level for a specific date and location |
| Primary Actors | TFL staff, App developers |
| Pre-conditions | User is logged in, user can access the machine learning app |
| Main flow | 1. User adds date and location parameters and clicks to make a prediction<br>2. System checks if user has appropriate permissions and if parameters are valid<br>3. If both are true, a prediction is made for pollution level for the corresponding parameters and displayed for the use. |
| Alternative flow | a. If user does not have permission, the user is prompted with a invalid permissions message<br>b. If user inputs invalid parameters, user is prompted with a re-enter parameters message |

# 4 References


[1] Business Analysis Body of Knowledge, “Elicitation,” BABOK Page, Jan. 20, 2015. https://babokpage.wordpress.com/elicitation/ (accessed Dec. 31, 2022).

[2] A. M. Hickey and A. M. Davis, “Elicitation technique selection: how do experts do it?,” Journal of Lightwave Technology, 2023, doi: 10.1109/icre.2003.1232748.

[3] A. Davis, O. Dieste, A. Hickey, N. Juristo, and A. M. Moreno, “Effectiveness of Requirements Elicitation Techniques: Empirical Results Derived from a Systematic Review,” 14th IEEE International Requirements Engineering Conference (RE’06), Sep. 2006, doi: 10.1109/re.2006.17.

[4] “Elicitation Techniques: Getting People to Talk About Ideas They Don’t Usually Talk About,” Theory & Research in Social Education, 2015. https://www.tandfonline.com/doi/abs/10.1080/00933104.2015.1034392 (accessed Jan. 02, 2023).

[5] O. Dieste and N. Juristo, “Systematic review and aggregation of empirical studies on elicitation techniques,” IEEE Transactions on Software Engineering, vol. 37, no. 2, pp. 283–304, Mar. 2011, doi: 10.1109/tse.2010.33.

[6] J. Nicholas, “9 Elicitation Techniques Used by Business Analysts – Tips and Guidance,” BusinessAnalystMentor.com, Dec. 10, 2021. https://businessanalystmentor.com/elicitation-technique/ (accessed Dec. 31, 2022).

[7] N. J. Cooke, “Varieties of knowledge elicitation techniques,” International Journal of Human-Computer Studies, vol. 41, no. 6, pp. 801–849, Dec. 1994, doi: 10.1006/ijhc.1994.1083.

[8] U. Rafiq, S. S. Bajwa, X. Wang, and I. Lunesu, “Requirements Elicitation Techniques Applied in Software Startups,” 2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Aug. 2017, doi: 10.1109/seaa.2017.73.

[9] C. Pacheco, I. García, and M. Reyes, “Requirements elicitation techniques: a systematic literature review based on the maturity of the techniques,” IET Software, vol. 12, no. 4, pp. 365–378, Aug. 2018, doi: 10.1049/iet-sen.2017.0144.

[10] D.-M. Vásquez-Bravo, M.-I. Sánchez-Segura, F. Medina-Domínguez, and A. Amescua, “Knowledge management acquisition improvement by using software engineering elicitation techniques,” Computers in Human Behavior, vol. 30, pp. 721–730, Jan. 2014, doi: 10.1016/j.chb.2013.09.003.

[11] P. Weichbroth, “Facing the brainstorming theory. A case of requirements elicitation,” Econstor.eu, 2016, doi: http://hdl.handle.net/10419/173338.

[12] Software Testing Help, “Top 10 Most Common Requirements Elicitation Techniques,” Software Testing Help, Dec. 05, 2022. https://www.softwaretestinghelp.com/requirements-elicitation-techniques/ (accessed Dec. 31, 2022).

[13] S. Famuyide, Business Analyst Learnings, Mar. 16, 2015. https://www.businessanalystlearnings.com/ba-techniques/2015/3/16/tips-for-developing-questionnaires-surveys (accessed Dec. 31, 2022).

[14] O. Dieste and N. Juristo, “Systematic review and aggregation of empirical studies on elicitation techniques,” IEEE Transactions on Software Engineering, vol. 37, no. 2, pp. 283–304, Mar. 2011, doi: 10.1109/tse.2010.33.

[15] Mary Ann Morrison, “The BA Zone,” The BA Zone, Jul. 12, 2017. https://www.thebazone.com/bablog/2017/7/11/survey-says-elicitation-tool-1 (accessed Dec. 31, 2022).

[16] D. M. Berry and E. Kamsties, “Ambiguity in Requirements Specification,” Perspectives on Software Requirements, pp. 7–44, 2004, doi: 10.1007/978-1-4615-0465-8_2.

[17] K. Waters, “Prioritization using MoSCoW,” 2009. [Online]. Available: https://comp.anu.edu.au/courses/comp3120/local_docs/readings/Prioritization_using_MoSCoW_AllAboutAgile.pdf

[18] Government Digital Service, “Data protection,” GOV.UK, Nov. 15, 2011. https://www.gov.uk/data-protection (accessed Dec. 31, 2022).

[19] Nuclino, “A Guide to Functional Requirements (with Examples),” Nuclino.com, 2022. https://www.nuclino.com/articles/functional-requirements (accessed Dec. 31, 2022).

[20] AltexSoft, “Non-functional Requirements: Examples, Types, How to Approach,” AltexSoft, 2019. https://www.altexsoft.com/blog/non-functional-requirements/ (accessed Dec. 31, 2022).

[21] M. Khan, G. Habibullah, J. Gay, and Horkoff, “Non-Functional Requirements for Machine Learning: An Exploration of System Scope and Interest,” doi: 10.1145/3526073.3527589.

[22] W3, “Web Content Accessibility Guidelines (WCAG) 2.1,” W3.org, Jun. 05, 2018. https://www.w3.org/TR/WCAG21/ (accessed Dec. 31, 2022).

[23] P. Rome, “What are Non Functional Requirements — With Examples | Perforce Software,” Perforce Software, 2020. https://www.perforce.com/blog/alm/what-are-non-functional-requirements-examples (accessed Dec. 31, 2022).