**EVALUATION TECHNIQUES**

This section is key to evaluating the potential success of **Commuty**, as it focuses on measuring the effectiveness and user satisfaction with the app.

Additionally, before entering the evaluation phase, we will pose certain fundamental questions to better define the product evaluation process and ensure that decisions align with user needs and project objectives. Questions such as: Is ease of use more important, or are we aiming to provide accurate information? Where do the critical success factors lie? What novel techniques could be employed to effectively measure work performance?

Since we are applying these evaluation techniques throughout the entire development process, including the final phases when the product is completed, we will use them as a guide in each phase, ensuring a clear path for each step related to these evaluations. This approach serves as a set of preventive rules to avoid mistakes and guarantee the best final outcome for the user when all system components are integrated.

Once we have implemented these preventive rules, it will be necessary to obtain direct feedback from users to verify that what we have designed meets real expectations and needs. For this, we have applied:

* Heuristics at various stages of the development process
* Metrics to analyze the application's responses in terms of time and workload
* Scales to evaluate how the user reacts to the modifications implemented
* An iterative adjustment process based on the feedback received.

All of this will allow us to fine-tune the product before its final release, ensuring that it meets the quality and usability standards set by the market.

In the development of Commuty, the standards we intend to follow are important. While we are aware of our limitations in terms of knowledge, time, and available tools, we are committed to meeting achievable criteria for us, such as app usability, resource optimization, and accessibility criteria. However, more demanding standards, such as the ability to handle high traffic, supporting at least 100,000 concurrent users without performance drop as proposed by Amazon Web Services (2021)**[16]**, would be out of our reach, since the TFL API under our current conditions (without requesting a special agreement with TFL) allows us to handle 500 requests per minute (TFL, 2025)**[2]**. That said, we have set goals to achieve fast load times, ideally under 3 seconds (Google, 2016)**[15]**, and low response times, preferably under 1 second for notifications (Preece et al., 2015)**[10]**. These aspects would be fundamental to providing a smooth, efficient, and satisfying experience for users, ensuring that the app is intuitive, accessible for everyone, and capable of operating efficiently even under high-demand conditions.

**HEURISTICS**

Heuristics are standardized usability principles that help evaluate the ease of use of Commuty. We will use **Nielsen's 10 heuristics** for evaluating the user interface (UI) and user experience (UX). But before diving into that, we need to evaluate the question: ***Is usability more important, or are we aiming to provide information?***

According to the concept and development of the project, the focus of Commuty is more on the **usage** of the app to **enhance the user experience during their daily commutes**. The main goal is to **make the user's life more efficient** by notifying them of disruptions in their usual route and offering alternatives so they can make decisions.

Although the **information** provided (delays or alternative routes) is crucial, **usability** (the ease with which the user can receive and act on that information) will be the most important factor. Based on this, we could conclude that heuristics related to the information given to the user, such as "***Visibility of system status***" or "***Help and documentation***", will be less critical for our project than others like "***User control and freedom***", "***Error prevention***" or "***Flexibility and efficiency of use***" which are more related to usability.

**EXPLANATION OF HEURISTICS:** Let’s explain each one and see if they are more related to usability or information, which will also help us determine if they are more or less critical for our app:

**1. Visibility of system status:** This focuses on the system providing clear and timely information to the user about the status of their interaction. Clear and proactive notifications are an effective way of keeping users informed about possible delays, interruptions, and alternative routes before they affect their journey. **Specific Example:**

* **Advance delay notification:** If the system detects a significant delay in the user's route, it shows a notification like "Victoria line has a 10-minute delay. Suggested alternative route."
* **Visual activity indicators:** While the app is checking the route, the "View route" button could change to an icon indicating that the check is in progress (e.g., a spinning circle), providing visual feedback about the process.

**Standards:** The **ISO 9241-210** standard states that applications must provide clear and consistent feedback, ensuring that users stay informed (ISO, 2010)**[1]**.

**2. Match between the system and the real world:** The system should speak the users' language, using terms and concepts that are familiar, instead of technical jargon. Additionally, icons, buttons, and text should be intuitive.

**Specific Example:**

* **Common language:** Instead of displaying technical terms like "Traffic API," the app could use a message like: "Checking real-time traffic" or "Searching for faster routes."
* **Intuitive button descriptions:** Instead of using an abstract "menu" icon, use clear text that says "View alternative routes" so that the user knows exactly what to expect when clicking.

**Standards:** According to the **Nielsen Norman Group**, using clear and accessible language improves usability, making the system more understandable for the end user (Nielsen, 1995)**[2]**.

**3. User control and freedom:** Users should be able to easily undo or redo actions. The system should allow users to feel in control of the app.

**Specific Example:**

* **Undo actions:** If the user selects an alternative route and then realizes they don’t want to take it, they can see a button that says "Cancel route change" or "Return to original route." This gives the user full control and freedom to reverse decisions.
* **Cancel notifications:** The user can access the app’s settings and turn off notifications for delays or interruptions on a specific route, giving them the freedom to personalize their experience.

**Standards:** The **ISO 9241-210 usability** standard recommends that interfaces allow users to easily correct errors and feel they have control (ISO, 2010)**[1]**.

**4. Consistency and standards:** The system should follow established conventions and standards, ensuring that behavior and design are predictable, as this makes the app easier to use.

**Specific Example:**

* **Uniform colors and style:** If a transport delay is shown in red in one place, that same color should be used on all screens that display route status. The same applies to icons and buttons. If an "Accept" button is green on one screen, it should remain green on all screens to maintain consistency.
* **Logical task flow:** The interface should follow a logical sequence of steps: first, set up the route, then show route options, set different preferences for notifications, and finally, store the information.

**Standards:** The **Nielsen Norman Group** emphasizes the importance of consistency in design to minimize cognitive load and improve usability (Nielsen, 1995)**[2]**.

**5. Error Prevention:** Help users avoid problems and reduce cognitive load. When alternatives or navigation options are clear and well-structured, the likelihood of making mistakes decreases. Additionally, if there is an input error, the app should be clear and tell exactly what is wrong and how to correct it.

**Specific Example:**

* **Input validation:** If the user enters an invalid time or date, the app should display a message like "The entered time is not valid. Please enter a time between 6:00 am and 9:00 pm." This prevents errors in data entry.

**Standards:** According to the **Nielsen Norman Group**, error prevention is essential to reduce user frustration and improve interface efficiency (Nielsen, 1995)**[2]**.

**6. Recognition rather than recall:** The system should minimize the user's cognitive load by displaying options and elements visibly so that the user can easily select them, instead of asking them to recall information from previous screens.

**Specific Example:**

* **Route setup screen:** When entering the "Route Setup" screen, the user sees a clear list of available options to enter their departure and destination points, with easily identifiable buttons to modify, add, or choose routes, without having to remember previous settings. Example: "Enter your starting point and destination. Where are you coming from today? Where are you headed?"
* **Alternative routes screen when there are delays:** If a problem with the route is detected, the app clearly shows two or more alternative routes on the notifications screen, without the user having to remember what options were displayed previously. Example: "The Victoria line has a delay. Here are two alternatives: 1) Bakerloo line to Oxford Circus, 2) Central line to Tottenham Court Road."

**Standards:** The **Nielsen Norman Group** recommends that apps clearly and accessibly display available options to avoid the user's cognitive load, ensuring that the flow of information is simple and direct (Nielsen, 1995)**[2]**.

**7. Flexibility and efficiency of use:** The app should adapt the user experience to be both efficient and **easy to use**. This is related to ease of use for beginners and **effectiveness** for more experienced users, as it allows shortcuts or customization.

**Specific Example:**

* **Novice and expert users:** For novices, the app shows a simple process for setting up the route and notifications. For more experienced users, they can enable an advanced option in the settings that allows them to choose the exact conditions under which they will receive notifications, such as "Only if the delay is greater than 10 minutes."
* **Quick access to settings:** If a user is familiar with the app, they should be able to quickly switch between the default route and an alternative, without navigating through too many menus or screens. Quick access to these functions from the main screen would be ideal.

**Standards:** **Nielsen (1995)** states that flexibility in design is crucial to accommodate both beginner and expert users, allowing the experience to be personalized according to skill level.**[2]**

**8. Aesthetic and minimalist design:** By applying a simple and clean design, focusing on what’s essential and eliminating information overload, both **usability** and **communication effectiveness** are improved. Relevant information is clearly displayed without distractions.

**Specific Example:**

* **Clean interface:** The main screen should only display the most relevant information for the user, such as the current route status and buttons to view alternatives. The colors and interface elements should be simple and well-organized to avoid distractions.
* **Clear and to-the-point notifications:** When there’s a delay or major change, the notifications should be brief but informative. Example: "10-minute delay on the Victoria line. Alternative route available."

**Standards:** The **ISO 9241-210** standard emphasizes the importance of a clean and simplified design that favors clarity and efficiency, eliminating any unnecessary information that could distract the user (ISO, 2010)**[1]**.

**9. Help users recognize, diagnose, and recover from errors:** If an error occurs, the system should provide clear messages indicating the problem and how to fix it.

**Specific Example:**

* **Clear error messages:** If the app cannot retrieve real-time data about the route status, it should display a clear message like "Unable to check route status. Please try again later." It should also offer suggestions or options to check later.
* **Proactive correction:** If the user sets a route with incorrect data (e.g., selects a past time), the app should display an error message and give clear instructions to correct the problem, such as "The selected time is in the past. Please choose a future time."

**Standards:** The **Nielsen Norman Group** highlights that error correction should be proactive, and apps should provide clear, action-oriented messages to help users resolve any issues quickly (Nielsen, 1995)**[2]**.

**10. Help and documentation:** Although the system should be intuitive enough that users do not need to read manuals, help should be available if needed.

**Specific Example:**

* **Onboarding tutorials:** When opening the app for the first time, a quick tutorial should guide the user on how to set up their first route, how to modify it, and how to manage traffic notifications.
* **Accessible help section:** Include a "Help" option in the menu that explains how the app works, how to set up custom alerts, and how to view route alternatives when there are disruptions. This could also include frequently asked questions (FAQs).

**Standards:** The **Nielsen Norman Group** also states that while the interface should be intuitive, it’s always helpful to provide an accessible help system for users who need assistance, making it easier to recover from any confusion or doubts (Nielsen, 1995)**[2]**.

**HEURISTIC EVALUATION:** After explaining with applicable examples for the app, we examine how well they are implemented by reviewing and evaluating the interface under the following parameters, an approach inspired by the recommendations from the article *How to Conduct a Heuristic Evaluation* (Moran & Gordon, 2023)**[5]** from the Nielsen Norman Group, though with an interpretation adapted to the specific needs of Commuty:

1. **When to Apply:** In this column, we specify the best point to apply each heuristic, grouped as follows:
   * **Initial Design and Prototyping Phase:** In this early phase, the primary purpose is to ensure that design concepts are effective and usable. For example, heuristics like "***Visibility of system status***", "***Match with the real world***" or "***Recognition rather than recall***".
   * **Development and Prototype Testing Phase:** In this phase, heuristics are applied to ensure that interactions are efficient and functional in the final product. Usability testing is done to identify issues and refine the design. Heuristics to apply in this phase would be "***Consistency and standards***", "***Error prevention***" or "***Flexibility and efficiency of use***".
   * **Evaluation and Launch Phase:** In this final phase, usability evaluations are performed to ensure that the app meets user expectations and that the experience is smooth, efficient, and free of critical errors. Heuristics like "***Simplicity and clarity***", "***Recognition rather than recall***" or "***Visual and aesthetic appeal***" belong in this phase.
2. **Evaluation Points:** For each heuristic, the evaluation points should focus on the key aspects that determine whether the interface meets usability expectations and design principles established by Nielsen. For each heuristic, two to three questions will be evaluated, adapted to each specific case. For example:
   * "*Is the language used understandable and accessible to the average user?*" for "***Match between the system and the real world***".
   * "*Can the user easily undo or modify their route?*" for "***User control and freedom***".
3. **Severity Rating:** While there is no exact score for each heuristic, the severity of the problem can be classified. Nielsen proposed a severity rating scale of **0 to 4** to rate the problems found (Nielsen, 1994)**[6]**:
   * **0:** **No problem** – Does not affect usability.
   * **1:** **Minor problem** – Low importance or barely noticeable.
   * **2:** **Minor issue** – A problem that could be improved without much difficulty.
   * **3:** **Major problem** – Negatively affects the user experience.
   * **4:** **Critical problem** – The interface is nearly unusable due to this issue.

Severity is measured based on three main factors:

* + **Frequency:** If it’s something frequent or will affect many users, the severity is higher.
  + **Impact:** If it significantly interferes with the task or prevents the user from completing their goals efficiently, it has a higher severity.
  + **Solubility:** If the problem has a simple and quick fix, its severity is lower, even if the impact is significant. If the solution is complex or requires a major redesign, the severity will be higher.

1. **Complies / Does not comply:** The evaluator determines whether the heuristic is met or not in the interface, typically with a "Yes/No." If it does not comply, the issue should be described.

| **Nielsen Heuristic** | **When to apply** | **Evaluation Points** | **Severity (0-4)** | **Meets? (Yes/No)** |
| --- | --- | --- | --- | --- |
| System Status Visibility | Prototyping and Development phases | * Are the delay and route change notifications clear? * Are activity indicators (such as loading icons or timers) easy to understand? * Does the system provide real-time updates? |  |  |
| Match with the Real World | UI/UX phase | * Is the language used understandable and accessible to the average user? * Are the buttons and options intuitive and easy to understand? |  |  |
| User Control and Freedom | Design and Prototyping phase | * Can the user easily undo or modify their route? * Are there clear options for changing notification settings? |  |  |
| Consistency and Standards | Design and Prototyping phase | * Is there consistency in the colors and styles for delay notifications? * Does the task flow follow a logical and predictable sequence? |  |  |
| Error Prevention | Prototyping phase | * Does the app proactively alert about potential delays before the user starts their journey? * Is the system clear when an input error occurs? |  |  |
| Recognition vs Recall | Prototyping phase | * Does the app present route options and alternatives clearly, without requiring the user to remember previous selections or information? * Does the user need to remember details, or is everything visible? |  |  |
| Flexibility and Efficiency of Use | Design and Prototyping phase | * Do novice users find it easy to set up their route, while experts can make quick adjustments? * Are there advanced options for more experienced users? |  |  |
| Aesthetic and Minimalist Design | Before and during prototyping | * Does the app display only the most relevant information for the user? * Is the interface free from unnecessary elements that distract the user? |  |  |
| Help Users Recognize, Diagnose, and Recover Errors | Design and Prototyping phase | * Does the app show clear messages when an error occurs? * Does the system offer easy-to-understand and apply solutions? |  |  |
| Help and Documentation | From the beginning, during the design phase | * Does the app provide contextual help when needed? * Are there tutorials or guides that are easy to find and use? |  |  |

**CRITICAL SUCCESS FACTORS**

These are the key elements that determine the feasibility and success of an application. These factors represent the essential conditions that must be met for the application to achieve its objectives, meet user expectations, and maintain adequate performance in the market. In the case of **Commuty**, they include the **accuracy and reliability of notifications**, the **relevance of alternative routes**, and **ease of use and accessibility**:

**1. Accuracy and reliability of notifications:** The app must notify users about delays in real-time, accurately and promptly. If this doesn't happen, users will lose confidence and may stop using it. ***An inaccurate notification, such as alerting a delay that has already been resolved or failing to notify a delay that affects the user, could cause frustration or even lead to users abandoning the app.***

**Solution:** Integrate real-time data from TFL and provide updates. **How to ensure it**:

* **Reliable data source:** In the system architecture design phase, we decided to use ***TFL's Unified API*** as it provides access to real-time data services, allowing us to get updates on the public transport status (Transport for London, 2025)**[3]**.
* **Customizable notifications:** In the same phase, we introduced the option to configure notification frequency based on the user's preferences (e.g., receive notifications only when there are significant delays of more than 5 minutes).

**When to apply:** During the design of the app’s architecture, when integrating the TFL APIs. It is also important during real-time functionality testing.

**2. Relevance of alternative routes:** Alternative routes must be effective and offer solutions that enhance the user experience by providing useful options during service interruptions. ***The alternative routes should be efficient enough for the user to consider them as viable options when an issue occurs with their usual route.***

**Solution:** Analyze real-time traffic and offer customizable options that fit the user's preferences. **How to ensure it:**

* **Traffic and transport conditions analysis:** Thanks to the TFL API, we can use real-time data to evaluate congestion, estimated arrival times, and traffic conditions to provide alternative routes (Transport for London, 2025)**[3]**.
* **Evaluation of alternatives:** Whenever an issue occurs on the user's usual route, the app should clearly present alternative options to the user, evaluating and prioritizing those that will help the user arrive on time.
* **Customization of alternatives:** Allow users to customize what type of routes they prefer (e.g., avoiding line changes, shorter waiting times, or shorter routes), making the alternatives even more useful for their personal needs.

**When to apply:** During the integration of the TFL API that provides real-time traffic data. Also, during usability testing to see if users perceive the alternatives as valuable.

**3. Ease of use and accessibility:** The app's interface should be intuitive, easy to use, navigate, and understand, especially in urgent contexts where users need to make decisions quickly, even for those who are not tech-savvy. If users find it difficult to set up routes, understand notifications, or search for alternatives, they are likely to stop using it. ***Simplicity and clarity in interaction are essential for the app to be accessible to a wide range of users, including those less familiar with technology.***

**Solution:** Use a simple design, conduct usability tests, and provide tutorials and contextual help. **How to ensure it:**

* **Simple and clean design:** We based the design on minimalism, ensuring that the most relevant information (delay notifications or alternative routes) is easy to access and understand without excessive searching (ISO, 2010)**[1]**.
* **Clear user flow:** The app should function simply from the start. To achieve this, we created an intuitive user flow for entering regular routes and receiving notifications, minimizing the number of steps to complete key actions (Nielsen, 1995)**[2]**.
* **Usability testing:** We will conduct usability tests with users of varying technical levels to ensure that the interface is accessible, observing how they interact with the app and adjusting design elements based on the difficulties they encounter (Nielsen Norman Group, 2025)**[4]**.
* **Clear instructions and support:** We will provide in-app tutorials to help users set it up correctly, offering an easy-to-access help system in case they need assistance.

**When to apply:** Primarily in the design and prototyping phase of the user interface. Usability testing should be conducted before launching the final product to ensure all users can interact with the app without complications.

**CRITICAL EVALUATION**

We will conduct a detailed and reflective analysis of the potential difficulties and limitations we could face when implementing the key features of the application. The idea is to identify critical points in our current approach and reflect on how they could affect the user experience or the reliability of the system:

**1. Potential Issues, How to Handle Cases Where Alternatives Are Not Better Than the Original Route?:** Sometimes, the suggested alternative routes may not be better in terms of time or comfort, which can lead to frustration or confusion among users (Nielsen, 1993)**[8]**. **How to address this**:

* **Prioritize arrival time:** Above all, the app must prioritize an alternative that allows the user to arrive on time, despite possible delays on their usual route. According to Preece et al. (2015)**[10]**, users mainly value punctuality, which justifies the app prioritizing alternative routes that ensure the user reaches their destination in the shortest possible time. Additionally, if there is more than one viable alternative, the app should display options prioritizing the user’s preferences, something supported by Budiu (2015)**[7]**, who emphasizes the importance of personalizing the experience to improve user satisfaction.
* **Dynamic analysis:** The app’s request to the TFL API should compare not only the duration of the routes but also factors such as traffic or trip comfort (if the data is available). This is a recommendation based on system design principles that optimize the user experience (Nielsen, 1994)**[6]**.
* **Transparency:** The app should clearly explain why it is suggesting an alternative route. This could include a brief explanation such as, “Alternative routes recommended to avoid disruption on your usual route.” Budiu (2015)**[7]** emphasizes that transparency in communication within an interface is crucial for maintaining user trust, especially when offering critical decisions such as choosing an alternative route.
* **Option to disable alternatives:** For frequent users, we included an option to disable alternative route notifications based on delays to their usual route, especially if they prefer not to change unless strictly necessary. Personalizing the user experience plays a key role in their satisfaction, as explained by Nielsen & Molich (1990)**[9]**, who highlight that allowing users to adjust the app’s features according to their preferences significantly contributes to the positive perception of the system.

**2. Implementation Challenges, Is It Technically Difficult to Implement Real-Time Notifications?:** Real-time notifications require seamless integration with live data systems from TFL and transportation traffic. Maintaining the reliability and timeliness of notifications in such a dynamic environment can be a technical challenge. **How to address this**:

* **Notification technology choice:** We have chosen to implement push notifications in Commuty due to their ability to send instant, personalized messages to users even when the app is not in use. This feature allows us to keep users informed about updates, special offers, and relevant events, enhancing the user experience and promoting retention. Moreover, push notifications are an effective tool to increase user engagement and promote specific actions within the app (LogRocket, 2025)**[18]**.
* **Continuous data integration:** The app integrates data from TFL through its API, which updates information constantly and accurately. We are aware that the TFL API has limitations regarding the frequency of requests per minute, which could affect the reliability of notifications (TFL, 2025)**[3]**.
* **Performance monitoring:** We must have a monitoring system to detect potential notification failures and be able to correct them.
* **Simulation and testing:** We will conduct simulations of different scenarios to see how the app behaves under varying conditions. This also includes testing different types of alternative routes to ensure they are effective.

**METRICS**

These are key indicators that will help us evaluate if **Commuty** is achieving its goals and providing a positive user experience. They can be combined to get a comprehensive view of the app's performance and effectiveness:

**1. Response Time:** This is the time the app takes to provide a notification about a delay or suggest alternative routes. A quick response is critical for the app to be useful in real-time. We need to measure the time between when a delay or service disruption occurs and when the app notifies the user. Ideally, notifications should be sent in real-time or with minimal delay, so the user can make informed decisions without wasting time. According to **ISO 9241-210 usability standards[1]**, an app's ability to provide information quickly and efficiently improves **user satisfaction** and **reduces frustration** (ISO, 2010)**[1]**. Furthermore, Nielsen (1993)**[8]** states that **response times should be under 5 seconds** to ensure users do not perceive the app as slow or ineffective. This means the time between the event (service delay) and the notification sent to the user should be minimal, ensuring users can act before delays impact their experience.

**2. Alternative Route Success Rate:** This metric measures the percentage of times the app suggests an effective alternative route that is more convenient than the original route to avoid potential delays in case of incidents. This is one of the most important metrics, as it determines whether the app really helps users avoid problems. Nielsen & Molich (1990)**[9]** assert that an efficient system must offer **viable alternatives** for critical tasks without creating confusion, allowing the user to make quick, well-informed decisions. The success rate in this context can be measured by calculating how many times the suggested alternatives are faster or more comfortable than the original route. This metric is key to ensuring that the app is fulfilling its purpose of helping users make informed decisions during their commutes.

**SCALES**

We should use them to obtain a more accurate measurement of user perception and satisfaction with **Commuty**. One of the most common and effective tools is the **System Usability Scale (SUS)**, which has been widely adopted to measure usability in applications and interactive systems (Brooke, 1996)**[11]**. This scale provides a clear view of how users perceive the app interface in terms of ease of use and overall satisfaction.

The SUS consists of 10 questions answered on a numeric scale from 1 to 5 (where 1 means "strongly disagree" and 5 means "strongly agree"), allowing us to capture a quantifiable and easily interpretable perception of the user experience (Sauro & Lewis, 2012)**[12]**.

**SUS Score Calculation:** Once users have completed the questions, their responses can be summed to obtain an overall SUS score, which typically ranges from 0 to 100. According to the literature, a score higher than 68 is generally considered above average in terms of usability, although interpretations vary depending on the product’s context and user expectations (Tullis & Stetson, 2004)**[13]**.

For the case of **Commuty**, adapting the SUS to measure more specific aspects of the app, such as delays in transportation service alerts, is more appropriate. By adjusting the SUS questions to address key elements like the reliability of notifications and the usefulness of alternative routes, we can get a more precise measurement of the effectiveness of these features in the user experience (Lewis, 2018)**[14]**.

**TECHNIQUES/TOOLS**

In addition to standard techniques like usability testing and quantitative metrics, you could explore innovative techniques such as:

**1. User Satisfaction Surveys (SUS):** Throughout the different phases of the project, we will measure how users feel about the concept, app elements, and their experience with the app. We will use Google Forms to conduct brief surveys that provide feedback on our decisions. **Application:**

* **Concept:** Survey to assess whether the app’s concept is useful for users and if the proposed features meet their needs.
* **Structure:** After developing the low-fidelity prototype, we will use another survey to get feedback on the app’s structure and notifications, evaluating whether the screen organization and information hierarchy are intuitive.

**2. A/B Testing:** Compare different versions of the interface or notifications and see which performs better in terms of effectiveness and user satisfaction using tools like Google Optimize or Optimizely. **Application:**

* **Structure/Interface design:** We will compare three versions of the app’s interface to see which design is more efficient and satisfying for users in terms of interaction speed and clarity.
* **Delay notifications:** We will create two versions of the delay notification (one more detailed with additional information like estimated resolution time and another shorter one) to see which is more effective in terms of user understanding and satisfaction.
* **Alternative route success rate:** To calculate the success rate of alternative routes, we will compare how much time it would have taken the user on their original route when there’s a delay, with how much time it would take using the alternative route suggested by the app. If the suggested alternative turns out to be less efficient or doesn’t resolve the issue, it will be considered a "failure" in the app’s suggestion.

**3. Real-Time Data Analysis and User Logs:** During high-fidelity prototype testing, we will measure how users navigate the app with the goal of identifying friction points, difficulties, or ineffective navigation paths using user behavior analysis tools like Google Analytics or Hotjar, which allow us to record clicks and navigation patterns. **Application:**

* **Interaction mapping:** We will use user journey analysis tools (such as Hotjar or Google Analytics) to observe how users navigate through the app’s screens. This will help identify steps where users stop or get confused, allowing us to improve these points.
* **Click and scroll tracking:** We will track which buttons, links, or screens are most frequently selected or ignored, which can show which features of the app are most useful or underused. For example, if many users abandon the app after seeing delay notifications, it could indicate that the alternatives offered are not attractive enough.
* **Interaction times:** We will measure how much time users spend in certain sections of the app, such as when selecting an alternative route. If users spend too much time on a screen, it may be a sign that the information is difficult to understand or the interface is not optimized enough.

**4. Real-World Scenario Simulation:** We need to see how the app responds in hypothetical situations where different types of interruptions or service issues are simulated, and how it responds to these events in real-time, analyzing the speed and relevance of the proposed alternatives. For this, we will use performance monitoring tools like Firebase Performance Monitoring or LoadRunner to identify bottlenecks in the app and improve the user experience. **Application:**

* **Interruption scenarios:** We will simulate various transportation service interruptions, such as a delay on the subway line or station closures, to analyze how the app responds.
* **App failure simulation:** We will test how the app handles failures or connection drops, which is important to ensure users still receive alerts and updates properly, even if the connection is intermittent.
* **Decision evaluation:** It’s also important to create scenarios with multiple route alternatives, checking how the user interacts with the app to choose the most appropriate one and ensuring that the system provides enough details to make quick and well-informed decisions.

**5. Load Testing:** We also need to observe how the app behaves under high traffic or usage conditions and measure how many users the app can handle simultaneously without losing performance. For this, we will use tools like Apache JMeter, which can simulate multiple users interacting with the app at the same time. **Application:**

* **Simultaneous user scenario:** We will simulate situations with a large number of users accessing the app at the same time to measure how the app performs when they try to perform actions such as searching for routes, receiving delay notifications, or accessing personalization options simultaneously.
* **Backend infrastructure evaluation:** In addition to evaluating performance within the app itself, we will test how the backend handles multiple concurrent requests from the app to the TFL API, helping us identify if the API infrastructure can handle high traffic without failures or crashes.

**REFERENCES**

1. ISO, (2010). **ISO 9241-210: Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems**. [online] Available at: <https://cdn.standards.iteh.ai/samples/77520/8cac787a9e1549e1a7ffa0171dfa33e0/ISO-9241-210-2019.pdf> [Accessed 09 Feb. 2025].
2. Nielsen, J. (1995). **Usability Engineering**. San Diego: Academic Press.
3. Transport for London(TFL). (2025). **Unified API documentation**. [online] Available at: <https://tfl.gov.uk/info-for/open-data-users/api-documentation> [Accessed 31 Jan. 2025].
4. Nielsen Norman Group, (2025). **Usability Heuristics for User Interface Design**. [online] Available at: <https://www.nngroup.com/articles/ten-usability-heuristics/> [Accessed 28 Jan. 2025].
5. Nielsen Norman Group (2025). *How to Conduct a Heuristic Evaluation*. [online] Available at: <https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/> [Accessed 28 Jan. 2025].
6. Nielsen, J. (1994). *Heuristic evaluation*. In Nielsen, J., & Mack, R.L. (Eds.), *Usability Inspection Methods*. John Wiley & Sons, New York, NY.
7. Budiu, R. (2015). *Mobile usability*. New Riders.
8. Nielsen, J. (1993). *Usability Engineering*. Academic Press.
9. Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. *Proceedings of the ACM CHI'90 Conference*, 249-256.
10. Preece, J., Rogers, Y., & Sharp, H. (2015). *Interaction design: Beyond human-computer interaction* (4th ed.). Wiley.
11. Brooke, J. (1996). *SUS - A quick and dirty usability scale*. Usability Evaluation in Industry, 189–194.
12. Sauro, J., & Lewis, J. R. (2012). *Practical Guide to the System Usability Scale: Background, Benchmarks & Best Practices*. San Francisco: Morgan Kaufmann.
13. Tullis, T., & Stetson, J. (2004). *A comparison of questionnaires for assessing web site usability*. Proceedings of the Usability Professionals Association Conference, 1–12.
14. Lewis, J. R. (2018). *The System Usability Scale: Past, Present, and Future*. International Journal of Human-Computer Interaction, 34(6), 577-590.
15. Google, 2016. *The need for mobile speed*. [online] Available at: <https://www.thinkwithgoogle.com/marketing-strategies/app-and-mobile/mobile-page-speed-conversion-data/> [Accessed 27 Jan. 2025].
16. Amazon Web Services, 2021. *Architecting for Reliable Scalability*. [online] Available at: <https://aws.amazon.com/blogs/architecture/architecting-for-reliable-scalability/> [Accessed 27 Jan. 2025].
17. Amazon Web Services, 2021. *AWS Auto Scaling: Setup, Best Practices, Tips*. [online] Available at: <https://awsforengineers.com/blog/aws-auto-scaling-setup-best-practices-tips/> [Accessed 27 Jan. 2025].
18. LogRocket, 2024. React Native push notifications: A complete how-to guide. [online] Available at: <https://blog.logrocket.com/react-native-push-notifications-complete-guide/> [Accessed 10 Feb. 2025].

**SURVEYS**

**SUS about the concept**

1. **"How much time does your daily commute from home to work or school approximately take?"**
   * **Scale**: Less than 30 minutes, 30-60 minutes, 1-1.5 hours, 1.5-2, More than 2 hours)
   * **Objective**: To understand the average daily commute time of users, which may influence the importance of route optimization.
2. **"How many transport changes (bus, metro, train, etc.) do you approximately make during your daily commute?"**
   * **Scale**: None, 1, 2, 3 or more
   * **Objective**: To gather information about how many transport changes users need to make, which can help identify points where problems (e.g., delays in connections) might arise.
3. **"How often do you experience unexpected delays or disruptions during your daily commute to work or school?"**
   * **Scale:** 1: Never, 2: Quarterly, 3: Monthly, 4: Weekly, 5: Daily/Very often
   * **Objective**: To understand the frequency with which users face delays or disruptions, which could make your app concept relevant.
4. **"What factors do you consider most important when choosing your daily route (e.g., travel time, cost, comfort, etc.)?"**
   * **Scale:** Options: Time, Cost, Comfort, Ease of use, Transport changes
   * **Objective**: To identify the users' priorities when choosing a route and if these align with the route optimization features based on preferences.
5. **"Once you are used to your daily route, how often do you check the traffic conditions before leaving home?"**
   * **Scale**: 1: Never, 2: Occasionally, 3: Only when I hear of a problem, 4: Only when I expect major delays, 5: I always check before leaving
   * **Objective**: To understand if users stop checking traffic status once they memorize their route and how much attention they pay to potential issues, which could highlight the need for proactive notifications or automated alerts.
6. **"Would you like to receive information about traffic problems or interruptions before they occur, so you can make decisions about your route?"**
   * **Scale**: 1: No, never, 5: Yes, definitely
   * **Objective**: To assess if there's interest in the proactive notifications feature and if users find it useful to get information before problems occur.
7. **"Would you feel comfortable trusting an app to help you manage the planning of your daily routes more efficiently, notifying you about problems and offering alternative routes?"**
   * **Scale:** 1: Not comfortable, 5: Very comfortable
   * **Objective**: To assess users' willingness to rely on technology to manage their routes and receive automated notifications. (If not comfortable, could this be due to privacy concerns? Maybe ask an additional question about this?)
8. **"Would you like to configure your journey once and not have to check traffic conditions every day, or do you prefer to personalize your route each day?"**
   * **Scale:** 1: Personalize each day, 5: Configure only once
   * **Objective**: To assess interest in automating the planning process where users configure their route once, and the app follows up and optimizes it daily or personalizes it each day.
9. **"How easy do you think it would be for you to use an app that helps you manage your daily route without requiring much interaction?"**
   * **Scale:** 1: Very difficult, 5: Very easy
   * **Objective**: To assess the perception of ease of use and simplicity of the app.
10. **"Do you think it’s convenient to receive information about your journey only when necessary, rather than receiving constant updates?"**
    * **Scale:** 1: Receiving constant updates, 5: Only when is necessary
    * **Objective:** To evaluate if users have concerns about notification frequency or if they prefer to receive alerts only when really necessary.

**SUS structure (Low-fidelity survey)**

1. **"It is easy to identify the key features of the app just by looking at the screens."**

* **Scale**: 1 (Very Difficult) to 5 (Very Easy)
* **Objective**: Measure how clearly the main functionalities are presented through the app’s visual design.

1. **"The layout of elements (such as buttons, input fields, and menus) on the screens makes it easy to visualize the user flow."**

* **Scale**: 1 (Very Difficult) to 5 (Very Easy)
* **Objective**: Evaluate if the layout of the elements facilitates smooth and logical navigation.

1. **"The design of the screens makes it easy to identify the most important options for my experience as a user."**

* **Scale**: 1 (Very Difficult) to 5 (Very Easy)
* **Objective**: Measure if users can quickly identify key and priority options within the app.

1. **"The images shown allow me to visualize how I could interact with the app in a simple way."**

* **Scale**: 1 (Very Difficult) to 5 (Very Easy)
* **Objective**: Assess if the images or mockups provided help users understand how to interact with the app.

1. **"I find it easy to understand how to input my schedules and routes based on the screenshots I saw."**

* **Scale**: 1 (Very Difficult) to 5 (Very Easy)
* **Objective**: Evaluate the ease of entering data (schedules and routes) based on what is shown in the prototypes.

1. **"The prototype screens clearly show how to view my daily routes."**

* **Scale**: 1 (Very Difficult) to 5 (Very Easy)
* **Objective**: Measure how clearly the prototype displays how to view daily routes.

1. **"Modifying my daily routes seems easy and intuitive based on the prototype images."**

* **Scale**: 1 (Very Difficult) to 5 (Very Easy)
* **Objective**: Evaluate if the process of modifying routes is clear and accessible according to the prototype design.

1. **"The notifications about issues on the route (shown in the prototype) appear clear, useful, and relevant."**

* **Scale**: 1 (Very Difficult) to 5 (Very Easy)
* **Objective**: Measure the effectiveness and usefulness of notifications related to issues on the route.

1. **"It is easy to see how to change or adjust the notification settings (such as receiving alerts about disruptions or delays on the route)."**

* **Scale**: 1 (Very Difficult) to 5 (Very Easy)
* **Objective**: Evaluate if users can easily adjust notification settings.

1. **"It is clear to me how the system will offer alternative routes if there is an issue with my usual route."**

* **Scale**: 1 (Very Difficult) to 5 (Very Easy)
* **Objective**: Measure if the system clearly presents how it will offer alternative routes in case of disruptions.