Università degli Studi di Torino

Dipartimento di Informatica

Bachelor's Thesis in Computer Science

Application Performance Monitoring: the APDEX Score

Supervisor:

Prof.Ssa: Susanna Donatelli

Candidate:

Leoluca D'Atri

Academic Year 2023/2024

Should array indices start at 0 or 1?
My compromise of 0.5 was rejected without, I thought, proper consideration.
- Stan Kelly-Bootle

Abstract

This thesis explores Application Performance Monitoring (APM) with a focus on the Application Performance Index (APDEX) score, a key standard for measuring user satisfaction with software performance. Based on an internship experience at Alten, an international consulting firm, the thesis examines the use of Dynatrace, an APM tool that tracks key metrics such as response times, user interactions, and more. By creating custom dashboards, the project translated complex technical data into visual insights, making it accessible for clients without technical expertise to monitor application performance and user experience in real time. The focus is on designing dashboards tailored to clients' needs on user experience monitoring. Practical examples demonstrate how different custom solutions were adapted to different client requirements, enabling effective tracking of user satisfaction and application performance.

I declare that I am responsible for the content of the work I am submitting for the purpose of obtaining the degree, that I have not plagiarized in whole or in part the work produced by others, and that I have cited the original sources in accordance with the current regulations on plagiarism and copyright. I am also aware that if my declaration is found to be false, I may face legal penalties and my admission to the final examination may be denied.

Contents

Chapter 1

Introduction

1.1 Company Overview

I completed my internship at Alten, an international consulting firm. During the first week, I was introduced to the company through presentations, which also briefly covered the technology I would be working with throughout my time there.

According to the presentations, Alten specializes in innovation, technology, and engineering. Founded in France in 1988, the company operates in various sectors, including information technology, electronics, embedded software development, telecommunications networks, and engineering.

Since its establishment in Italy in 2004, Alten has expanded to include 12 locations, with core competencies in ICT consulting, technology consulting, solutions, telecommunications, and life sciences.

1.2 Internship Objectives

The focus of the internship was to learn how to use the Application Performance Monitoring (APM) tool, Dynatrace, to provide clients with infrastructure monitoring capabilities. The ultimate goal was to develop skills in creating dashboards within Dynatrace that offer various types of visualizations based on key performance indicators (KPIs) that are both relevant and important to the client.

A KPI, or Key Performance Indicator, is a measurable value that indicates how effectively a specific objective is being achieved. For example, a KPI in the context of my work could be the response time of a web application, which helps assess whether the application's performance meets the desired standards and can directly influence potential cost savings. For instance, identifying performance bottlenecks, like the response time, can lead to optimized resource allocation, reducing unnecessary expenses and improving overall efficiency.

I was assigned to Alten's APM team, which consisted of several employees, each responsible for a number of clients. The team provides monitoring services tailored to the specific needs of each client. For instance, if a client prefers to focus solely on monitoring a particular problem area, the team delivers a solution that addresses that specific request. As a result, the scope of services provided varies significantly and can cover the entire range of Dynatrace platform offerings, depending on the client's requirements.

The sub-division I worked with specialized in User Experience, focusing on generating routine reports in various formats for clients, with a particular emphasis on dashboards.

A significant portion of my time was dedicated to training through Dynatrace University (https://www.dynatrace.com/dynatrace-university/), where I completed several courses covering the primary features of Dynatrace. I then progressed to more intermediate-level training, focusing specifically on User Experience and dashboard creation.

1.3 Application Performance Monitoring

Application performance monitoring (APM) is used to track and analyze key performance metrics of software applications. By doing this, APM helps improve service response times, optimize system performance, and enhance the overall user experience. [1]

According to Gartner: "Application performance monitoring (APM) is a suite of monitoring software comprising digital experience monitoring (DEM), application discovery, tracing and diagnostics, and purpose-built artificial intelligence for IT operations.“ [2]

Maintaining an application's optimal performance is crucial. The link between a user's experience and the back-end services that support it can be unclear, particularly in distributed cloud applications. Using these tools makes this easier as it provides a centralised place to monitor things.

1.3.1 Monitoring or Management?

It is very likely that during research or work related to Application Performance Monitoring, one also comes across another term. APM also refers to application performance **management**. While application performance monitoring emphasizes specific metrics and measurements, application performance management encompasses the broader discipline of developing and overseeing an application performance strategy. Both terms relate to interconnected technologies and practices [1].

1.4 APDEX History

I will briefly introduce the most important key metric that I utilized throughout my work: the Application Performance Index (APDEX), providing the understanding necessary for a more detailed exploration in its dedicated chapter.

APDEX is an open standard designed to offer a uniform method for reporting and bench-marking application performance, regardless of the measurement technique used.

The origins of APDEX date back to 2004, when Peter Sevcik and a group of application performance measurement experts formalized this approach through the Apdex Technical Specification V1.0, establishing a standardized method for reporting on application performance irrespective of the measurement technique [3].

1.5 Dynatrace

Where Dynatrace stands in respect to their market position is best explained through Gartner's accredited research.

1.5.1 Gartner and Magic Quadrant

Gartner, Inc., founded in 1979 by Gideon Gartner, is a renowned American analyst and consulting firm [4] and is a member of the S&P 500. [5]

The Magic Quadrant (MQ) is a set of market research reports published by the IT consulting firm Gartner. These reports use proprietary qualitative data analysis methods to showcase trends in the market, such as direction, maturity, and participants. [6] Gartner conducts these analyses across various technology sectors, with updates typically released every 1-2 years, after which the previous version is considered "retired" [7].

Rating

Gartner evaluates the vendors based on two criteria: completeness of vision and ability to execute [6, 8].

Completeness of Vision: "Reflects the vendor's innovation, whether the vendor drives or follows the market, and if the vendor's view of how the market will develop matches Gartner's perspective.“ [8]

Ability to Execute: "Summarizes factors such as the vendor's financial viability, market responsiveness, product development, sales channels and customer base.“ [8]

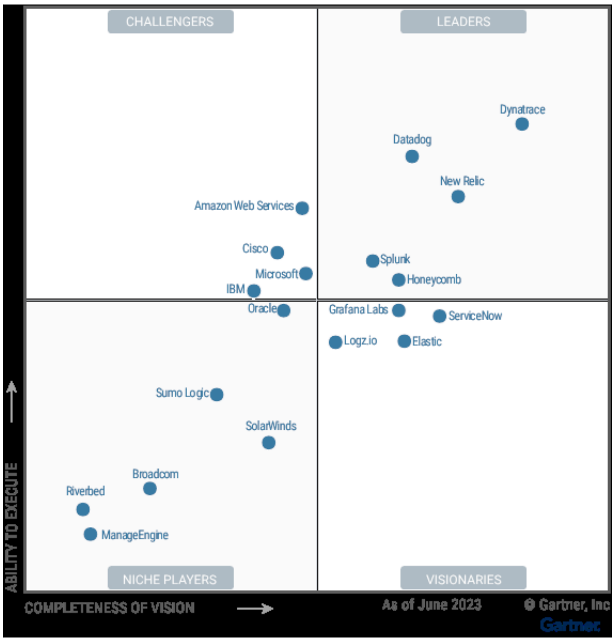


Figure 1.1: Gartner Magic Quadrant

Dynatrace is a Leader in this Magic Quadrant. Vendors in the "Leaders" quadrant have the highest composite scores for their aforementioned rating criteria.

1.5.2 Dynatrace

Dynatrace, Inc. is a global technology firm that offers a software observability platform based on artificial intelligence and automation. Its solutions are utilized for monitoring, analyzing, and optimizing application performance, software development and security practices, IT infrastructure, and **user experiences** across enterprises and governmental organizations worldwide [9, 10].

The Dynatrace platform utilizes a proprietary AI called **Davis** to discover, map, and monitor applications, microservices, container orchestration platforms (such as Kubernetes), and IT infrastructure across various network environments. The platform also provides automated problem remediation capabilities [11, 12, 13].

While the platform offers a broad range of features, my focus will be on its user experience and dashboard capabilities. The team I worked with oversees a wider array of the platform's features, but at times, smaller subsets of the team, such as myself in this case, focus on more specific aspects.

Chapter 2

Application Performance Index (APDEX)

The Application Performance Index, www.apdex.org, is an application performance industry specification proposed by the Apdex Alliance of companies as the Apdex Technical Specification [14]. According to it, the index is computed as follows:

$$A_T = \frac{C_s + 1/2C_t}{C_s + C_t + C_f} \tag{2.1}$$

A_T = Apdex Index: the overall score, ranging form 0 (no one is satisfied) to 1 (everyone is satisfied), summarising user satisfaction levels with application performance.

C_s = Count of Satisfied Responses: the number of interactions with the application that users perceive as quick and seamless, allowing them to focus entirely on their tasks without any disruption to their workflow.

C_t = Count of Tolerating Responses: the number of interactions that exceed the ideal response time but remain within a range that users can accept. These responses, while noticeable and potentially slightly disruptive to productivity, do not critically hinder the usability of the application. They are weighted at $\frac{1}{2}$ compared to satisfied responses.

C_f = Count of Frustrated Responses: the number of interactions with response times that are too slow to maintain a satisfactory user experience. In this range, users become dissatisfied; casual users may abandon their activities, while production users might interrupt their tasks.

The three zones are defined by two thresholds, T and F, in seconds, as follows:

- **Satisfied Zone**: 0 to T
- **Tolerating Zone**: $> T$ to F
- **Frustrated Zone**: $> F$

The value of F is four times the value of T. Assuming as an example that a user perceives a tolerable response time as 5 seconds, they would be frustrated at a response time of 20 seconds [15].

The threshold value T is defined by the organization using the formula. This value is chosen based on the organization's understanding of user expectations and the performance goals of their application. This threshold is set at the beginning of the monitoring period and can be adjusted as needed [14].

Figure 2.1 illustrates what I have explained visually.

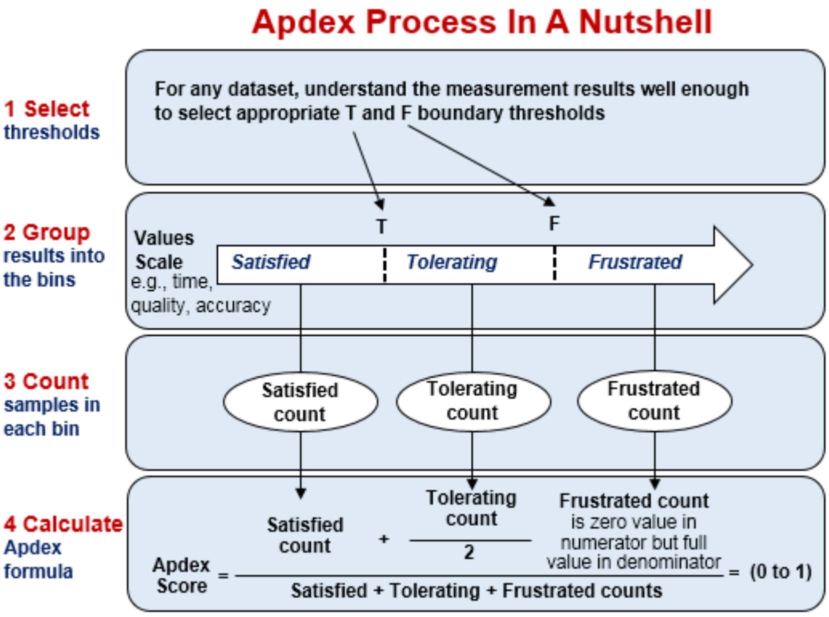

Apdex Process In A Nutshell
1 Select thresholds
For any dataset, understand the measurement results well enough to select appropriate T and F boundary thresholds
2 Group results into the bins
T
F
Values Scale e.g., time, quality, accuracy
Satisfied
Tolerating
Frustrated
3 Count samples in each bin
Satisfied count
Tolerating count
Frustrated count
4 Calculate Apdex formula
Apdex Score = (Satisfied count + Tolerating count / 2) / (Satisfied + Tolerating + Frustrated counts) = (0 to 1)
Frustrated count is zero value in numerator but full value in denominator

Figure 2.1: APDEX Process - www.apdex.org

Chapter 3

Dynatrace

3.1 History

I have based this section primarily on a video interview with Dynatrace founder and CTO, Bernd Greifeneder, titled “Disrupt or be Disrupted: The Story Behind the Dynatrace Reinvention,” published by Tech Seeking Human on YouTube in August 2023 [16].

In its early days, Dynatrace primarily catered to power users, offering a complex interface that wasn’t very user-friendly for the average customer. This realization led to a shift in how they handled and presented data. The company initially tried to integrate various monitoring tools into a unified solution, but this approach risked creating a disjointed, “Frankenstein” software product. This challenge eventually prompted Dynatrace to undertake a complete rewrite and re-imagine the platform from the ground up.

The rework was ambitious, focusing on starting from scratch and breaking down services into smaller components. This allowed for automatic detection of dependencies and identification of root causes for issues. The algorithm developed to support these features essentially resulted in what could be described as a “by definition causal AI.”

“Causal AI is an artificial intelligence technique used to determine the exact underlying causes and effects of events or behaviors.“ [17]

The rework was put into action by establishing a startup within Dynatrace, which concentrated on standalone SaaS offerings and emphasized innovation over enterprise concerns. This strategy, along with automation, led to the development of NoOps and a self-healing architecture, which streamlined operations and reduced the need for human intervention.

As the reworked Dynatrace gained traction, migration to the new platform commenced. However, the migration faced challenges in overcoming the natural human psychology’s resistance to change. Despite this, customers began to recognize its transformative benefits. Within two years, initial apprehension gave way to acceptance, and customers embraced the platform, acknowledging its impact. This acceptance spurred exponential growth and solidified Dynatrace into what it is today.

3.2 Functionality

This section is also supplemented by the training I received on the Dynatrace platform, alongside regular citations.

The Dynatrace unified observability and security platform uses AI to provide infrastructure monitoring, applications and microservices monitoring, application security, digital experience, business analytics, and cloud automation capabilities [18]. Dynatrace also provides the Dynatrace Hub, which publishes Dynatrace applications for specific observability use cases [19]. One of these applications, Dashboards, is the one I have used extensively my work.

Within dashboards, I only use features from one of Dynatrace's main features: **Digital Experience Monitoring.**

“Digital experience monitoring (DEM) is the practice of using tools and technologies to evaluate metrics from multiple sources that affect end users—such as applications, distributed cloud networks, user behavior, Internet of Things (IoT) devices, location-based data, and more—to determine the quality of a user's interaction with a digital touchpoint. DEM provides real-time metrics, reports, and alerts on key performance indicators (KPIs) that are important to your organization.“ [20]

Digital Experience Monitoring has two main components, Real User Monitoring (RUM) and Synthetic Monitoring [21]. Dynatrace also offers Session Replay, to replay exactly what users do in a session through generated video recordings and Mobile App Monitoring, to allow monitoring on native mobile applications [22].

Real User Monitoring (RUM)

Real User Monitoring is a technology that tracks and analyzes every interaction users have with applications. By gathering data directly from users' devices, it offers insights into performance from the user's point of view. RUM helps understand user behavior and assess application performance, while also identifying issues users encounter in real time [21].

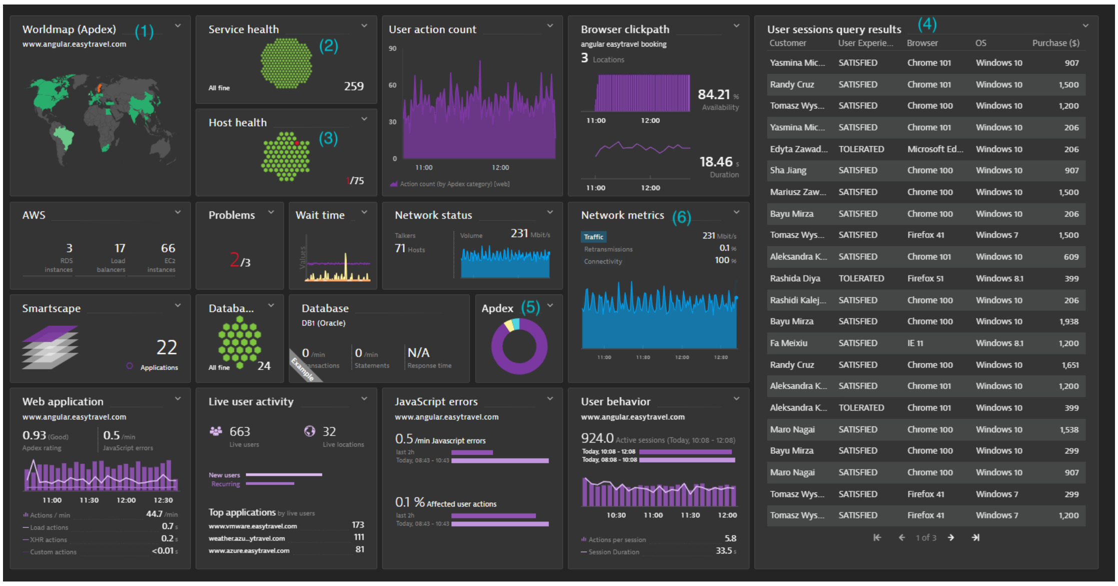
Synthetic Monitoring

According to the Dynatrace documentation, “Synthetic monitoring is about proactively simulating user visits“ [23]. In essence, it's a process of mimicking what a real user might do on your site, but performed by a robot. The same article mentions that there are different types of synthetic monitors available, though these are not relevant to my work.

Chapter 4

Monitoring Examples

The primary focus of my work is creating business-oriented dashboards. Based on my training, these dashboards are designed to aggregate and visualize key performance indicators (KPIs), metrics, and other critical data points that are relevant to an organization. By presenting data in a user-friendly format, such as dashboards, they make it easier for businesses to interpret valuable information, particularly those that may not have the time or resources to allocate for in-house monitoring solutions. This type of visualization ensures that businesses remain consistently informed about the status of their infrastructure.



User sessions query results (4)

Customer	User Experie...	Browser	OS	Purchase ($)
Yasmina Mic...	SATISFIED	Chrome 101	Windows 10	907
Randy Cruz	SATISFIED	Chrome 101	Windows 10	1,500
Tomasz Wys...	SATISFIED	Chrome 100	Windows 10	1,200
Yasmina Mic...	SATISFIED	Chrome 101	Windows 10	206
Edyta Zawad...	TOLERATED	Microsoft Ed...	Windows 10	206
Sha Jiang	SATISFIED	Chrome 100	Windows 10	907
Mariusz Zaw...	SATISFIED	Chrome 100	Windows 10	1,500
Bayu Mirza	SATISFIED	Chrome 100	Windows 10	206
Tomasz Wys...	SATISFIED	Firefox 41	Windows 7	1,500
Aleksandra K...	SATISFIED	Chrome 101	Windows 10	609
Rashida Diya	TOLERATED	Firefox 51	Windows 8.1	399
Rashidi Kalej...	SATISFIED	Chrome 100	Windows 10	206
Bayu Mirza	SATISFIED	Chrome 100	Windows 10	1,938
Fa Meixiu	SATISFIED	IE 11	Windows 8.1	1,200
Randy Cruz	SATISFIED	Chrome 100	Windows 10	1,651
Aleksandra K...	SATISFIED	Chrome 101	Windows 10	1,200
Aleksandra K...	TOLERATED	Chrome 101	Windows 10	399
Maro Nagai	SATISFIED	Chrome 100	Windows 10	1,538
Bayu Mirza	SATISFIED	Chrome 100	Windows 10	299
Maro Nagai	SATISFIED	Chrome 100	Windows 10	907
Tomasz Wys...	SATISFIED	Firefox 41	Windows 7	299
Tomasz Wys...	SATISFIED	Firefox 41	Windows 7	1,200

1 of 3

Figure 4.1: Example of a business dashboard

Figure 4.1 shows an example of a dashboard that represents the type of data a client is likely to find valuable. These dashboards are typically tailored to meet the specific requirements set by the client, such as tracking the global status of their APDEX score **(1)**, monitoring any downtime in hosts or services **(2)(3)**, or for example a detailed table that showcase user sessions, user experience, browser and OS data, and purchase amounts **(4)**. Many of these tiles recur in the dashboards I have created, as they pertain to Real User Monitoring.

Essentially, business dashboards translate technical data into actionable business insights. I will showcase these insights in more detail in the next chapter. They achieve this through various visualization techniques, such as charts **(5)**, graphs **(6)**, gauges (not present in the example), and more. These methods simplify complex datasets into information that is easy for clients to understand, particularly those who may not have technical expertise. This makes it easy for them to make well-informed business decisions.

4.1 Client Examples

I'm going over two different approaches used by the team I worked with at Alten to provide APM services in this section.

4.1.1 Example 1 - Report

The services the team provides to clients are typically customized to meet the specific needs of each client. One method of delivering important information is through a bi-weekly report—a text and media document that focuses on a specific application within a client's infrastructure.

The report covers several areas and, where applicable, includes direct links to the relevant sections of the Dynatrace instance. It starts with an overview of the Apdex score and a detailed analysis of user actions. Following this, it examines calls made to third-party services and analyzes specific failure rates, summarizing each request's failure reasons and error codes. It also categorizes median response times and failure rates by domain, along with their respective stack traces.

Problems and errors are summarized in a dedicated table, with one or two issues analyzed in greater depth. The report concludes with a Work In Progress (WIP) section, which is as a retrospective review of past activities and a summary of ongoing efforts.

4.1.2 Example 2 - Dashboard

This second method aligns closely with what I ended up working on and showcasing in the next chapter. This methodology consolidates all key data about the client's digital environment into one or more business-oriented dashboards.

The main dashboard covers frontend performance using Apdex scores over a defined period, with links to detailed Dynatrace pages for each web application. If an application's Apdex score is low, an additional dashboard offers more detailed analysis at the application level, linked directly alongside the main view. This dashboard also covers user demographics, service performance, error tracking, and user conversion paths, all tailored to the client's requests.

For each application-specific dashboard, the analysis is customized to fit the particular characteristics of that application. Commonly included metrics are user satisfaction by region, overall web application health, and the relationship between user frustration or bounce rates and application errors. Additionally, it can provide detailed insights into user navigation patterns when requested by the client.

In the following chapter, I'll explore this dashboard type in more detail and extensively explain its different aspects.

Chapter 5

Dashboards

As previously discussed in the preceding chapter, I was tasked with creating business-oriented dashboards that address a few key questions. These questions, based on the company's experience with various clients, were considered important to answer in order to facilitate informed business decisions

- How many users are satisfied according to the APDEX?
- What is the geographical distribution of users?
- Why and where did users exit the navigation?
  - Which parts of the website error out the most?
  - Is there a correlation between users that leave and those errors?
- How are users reaching the website?
- What activities do users perform the most on the website and how long does it take for them to be **visually complete**?
  - Are they *returning* users?

Let's first define some terms that I'll use during the dashboard review:

What are returning users?

"Returning users" are individuals who have previously visited an application and are recognized by the system during subsequent visits. This classification is based on cookies that help distinguish between first-time and returning visitors to a website or application.

What does "visually complete" mean?

According to Dynatrace, the term "visually complete" refers to the point in time when all the visible elements of a web page are fully displayed on the screen, regardless of whether background processes or non-visible elements are still loading.

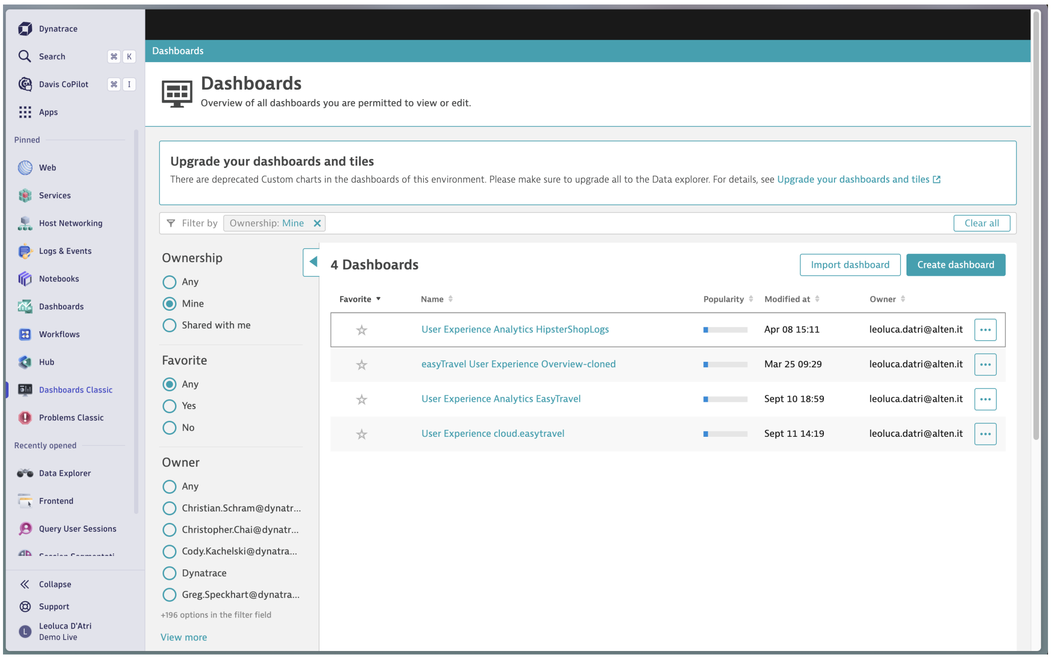
This metric is important in understanding user experience because it indicates the moment when a user perceives the page to be fully loaded, even if additional loading activities may still be occurring in the background.

To answer the questions, I was provided with a demo environment, a sandbox-like setup that allowed me to manipulate example infrastructure and analyze monitored data. This environment, given to interns at Alten, serves as a training ground to familiarize us with the Dynatrace platform. While the demo environment supports all Dynatrace features, I focused exclusively on user experience analysis.

I was tasked with answering the questions for two distinct infrastructures. The first was *easyTravel*, a comprehensive travel portal where users can log in, search for and select journeys, and book their travels. From a business perspective, *easyTravel* also includes a B2B portal for travel agencies to manage their offerings. The second infrastructure was a simpler one, *HipsterShopLogs*, an e-commerce website with basic functionality.

To be more specific, I was asked to monitor cloud.easytravel.com, rather than the easyTravel infrastructure. However, after consulting with the team lead, due to only allowing for restricted monitoring I was instructed to first create a dashboard for easyTravel and then do a few more specific things in a secondary dashboard for cloud.easytravel.com.

In order to create the dashboards, Dynatrace provides a dedicated Dashboard Page, as shown in Figure 5.1



Dashboards

Overview of all dashboards you are permitted to view or edit.

Upgrade your dashboards and tiles

There are deprecated Custom charts in the dashboards of this environment. Please make sure to upgrade all to the Data explorer. For details, see Upgrade your dashboards and tiles

Filter by Ownership: Mine ✕ Clear all

Ownership: ○ Any ● Mine ○ Shared with me

Favorite: ● Any ○ Yes ○ No

Owner: ○ Any ○ Christian.Schram@dynatr... ○ Christopher.Chai@dynatr... ○ Cody.Kachelski@dynatra... ○ Dynatrace ○ Greg.Speckhart@dynatra...

+196 options in the filter field

4 Dashboards

Favorite	Name	Popularity	Modified at	Owner	
☆	User Experience Analytics HipsterShopLogs		Apr 08 15:11	leoluca.datri@alten.it	···
☆	easyTravel User Experience Overview-cloned		Mar 25 09:29	leoluca.datri@alten.it	···
☆	User Experience Analytics EasyTravel		Sept 10 18:59	leoluca.datri@alten.it	···
☆	User Experience cloud.easytravel		Sept 11 14:19	leoluca.datri@alten.it	···

Figure 5.1: Dashboards Selection Screen

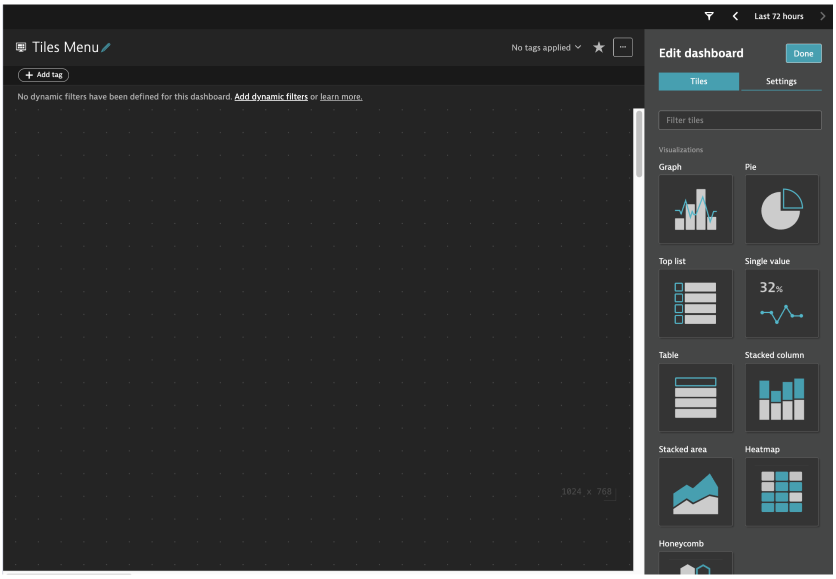
Dynatrace has recently undergone modernisation, porting old functionality to a new design. At the time of my internship, dashboards were not fully ported yet, and some specific functionality, such as certain types of tiles, still required the use of Dashboards Classic [24]. My manager also recommended using the classic variant of the dashboards, as it is the version they actively use with their clients.

According to Dynatrace's "Upgrade to Dashboards" support article [25], several features crucial to my dashboards are not yet portable to the new Dashboards app. These features fall under the category of "**built-in tiles**", a term for which I couldn't find a clear definition in their documentation.

Some of the built-in tiles that are listed and which I used are:

- Top web applications
- User behaviour
- World map
- Bounce rate
- JavaScript errors
- User Sessions Query – essential tile that I used extensively in my dashboards

Figure 5.2 shows how once created, the dashboard opens in an "edit" state, displaying a menu of available tiles, on the right-hand side of the screen.



No dynamic filters have been defined for this dashboard. Add dynamic filters or learn more.

Figure 5.2: New dashboard in edit state

The full selection of available tiles is illustrated in Figure 5.3:

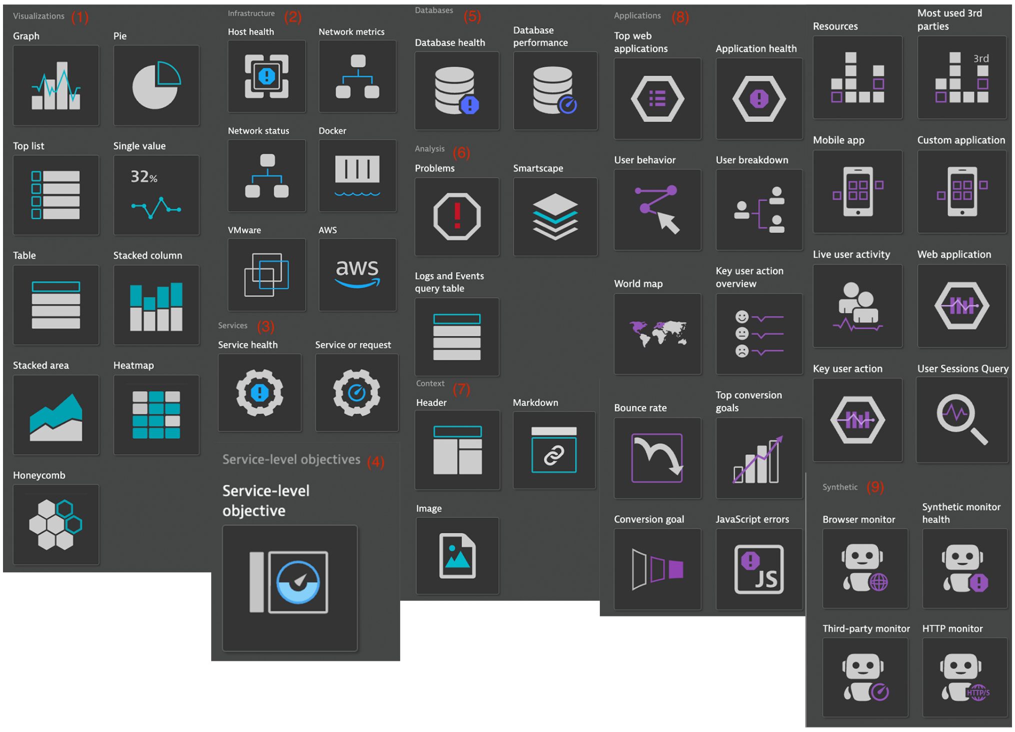


Figure 5.3: All dashboard tiles

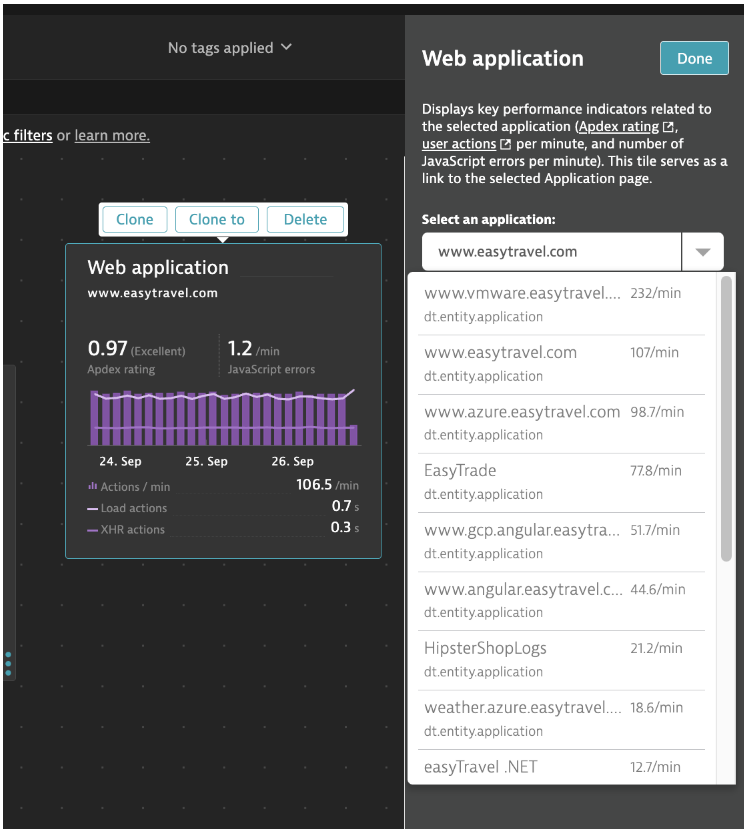
These tiles are highly versatile and provide a solid set of options. They are categorised into:

- Visualizations (1): Basic data representation tools
- Infrastructure (2): Host, network, and cloud service monitoring
- Services (3): Service health and request tracking
- Service-level objectives (4): Performance goal metrics
- Databases (5): Database health and performance monitoring
- Applications (6): Web and mobile app performance analysis
- Analysis (7): Problem detection and system visualization
- Context (8): Headers and rich text for additional information
- Synthetic (9): Proactive performance testing tools

Out of these, I've mainly had to use the ones from the Applications (6) section, the Visualizations section (1) and occasionally using Header and Markdown tiles (7).

Some tiles can be easily configured within the menu itself, allowing a few parameters to be set on the fly. Others require more extensive configuration through either the **Data Explorer** (typically tiles from the Visualizations category) or **User Sessions Queries** (USQ).

An example of a tile that doesn't require more specific configuration is Figure 5.4, "Web Application", from the Applications (6) section:



Displays key performance indicators related to the selected application (Apdex rating, user actions per minute, and number of JavaScript errors per minute). This tile serves as a link to the selected Application page.

Select an application: www.easytravel.com

www.vmware.easytravel.... dt.entity.application	232/min
www.easytravel.com dt.entity.application	107/min
www.azure.easytravel.com dt.entity.application	98.7/min
EasyTrade dt.entity.application	77.8/min
www.gcp.angular.easytra... dt.entity.application	51.7/min
www.angular.easytravel.c... dt.entity.application	44.6/min
HipsterShopLogs dt.entity.application	21.2/min
weather.azure.easytravel.... dt.entity.application	18.6/min
easyTravel .NET	12.7/min

Figure 5.4: Simple Tile Example

As it is a pre configured tile, all we have to tweak is the application it checks the data for, which can be done in the same sidebar menu where the list of tiles was.

Data Explorer and User Sessions Query The **Data Explorer** is a tool for querying and visualizing metrics. It is the backbone of the Visualizations section of tiles showcased earlier. Its core functionalities include:

- **Metric Querying:** Users can construct detailed queries using components such as metric names (identifiers for metrics like "CPU usage %", "Action count", etc.), aggregation methods, dimensions for splitting data, sorting options, rate settings, and filters.

- **Visualizations:** Multiple visualization types (e.g., graphs) can be used to represent the queried data. These can be integrated into dashboards for continuously updated monitoring, which is exactly what we use it for.

- **Baselining:** Provides baselining features to add confidence bands (boundaries) around metrics, making it easy to highlight deviations from expected patterns.

User Sessions Query (USQ) is very similar to Data Explorer but supports a more specific use case, which is illustrated in the section below. It is also its own tile, found under the Applications tile section.

During my internship, I questioned the difference between **Data Explorer** and **USQ**. According to a Dynatrace forum thread [26], **Data Explorer** is used to query and visualize data based on persistent time series information, which is stored in Cassandra. In contrast, USQs focus specifically on user session data, leveraging ElasticSearch for quick, session-specific insights, rather than persistent data over time. This allows USQs to answer questions directly related to user behavior, while Data Explorer addresses broader infrastructure and application metrics.

Both tools use Dynatrace's querying language (DDL) https://docs.dynatrace.com/docs/platform/grail/dynatrace-query-language, but the distinction in their underlying data sources means they serve different analytical needs. Understanding these differences helped me choose the appropriate tool for various analysis tasks during my internship.

I've made use of all of these tile types.

5.1 easyTravel

5.1.1 Primary Dashboard Overview

Frontend Overview

This section of the dashboard is designed to give a clear overview of the overall user experience across the entire easyTravel infrastructure. This makes it easier to quickly identify satisfaction trends and areas that may require improvement

I'll be reviewing the questions that I need to answer after each dashboard section, referencing the tiles that answer them directly. Some sections are reviewed together for better understanding.

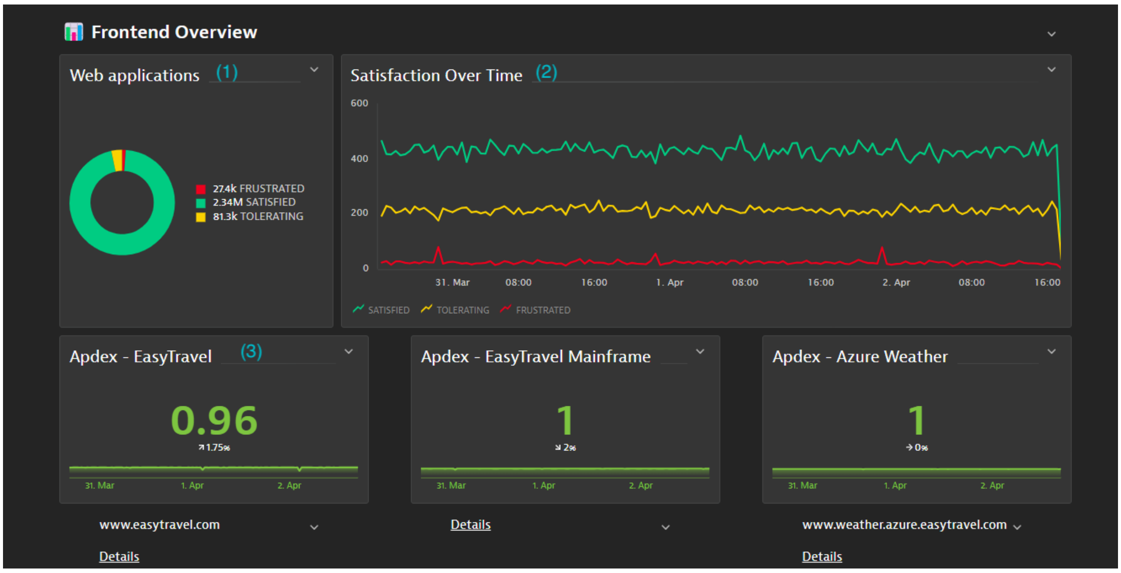


Figure 5.5: Dashboard 1 Frontend Overview. Annotation (3) is repeated for each APDEX tile shown.

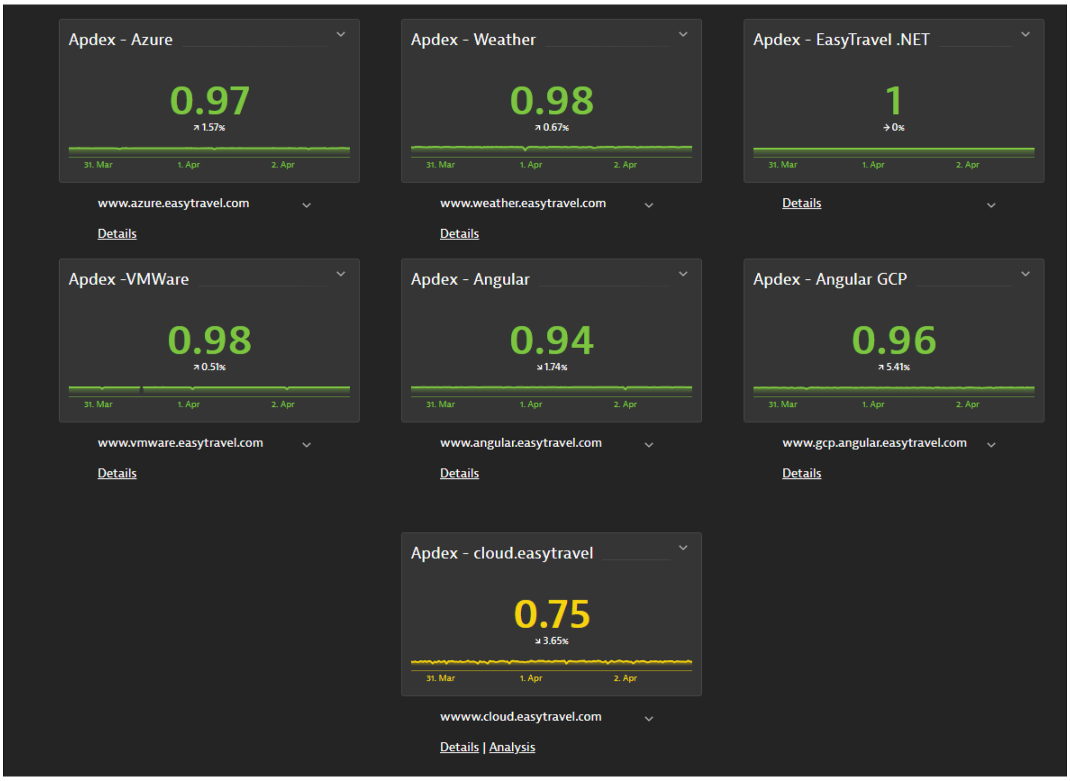
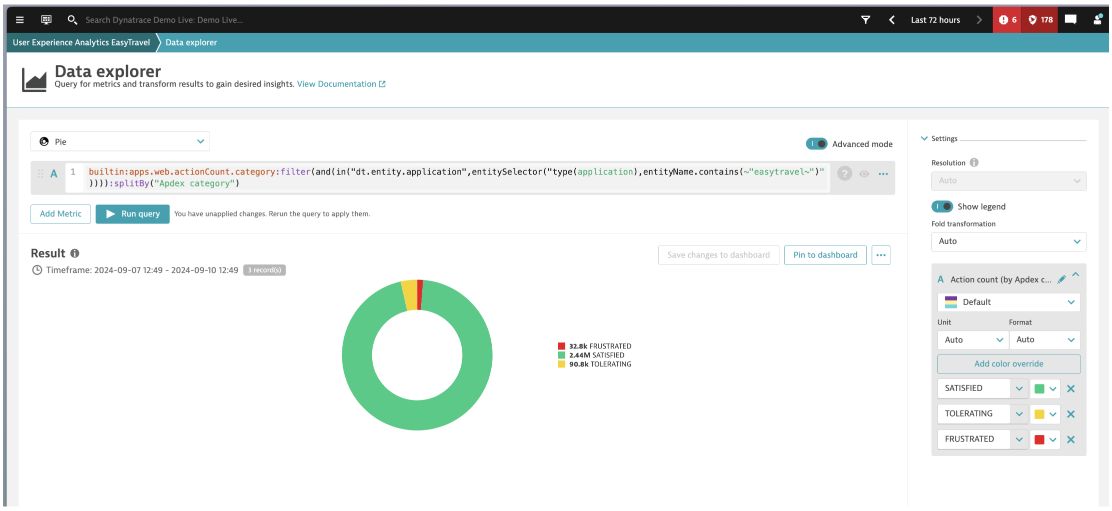


Figure 5.6: Dashboard 1 Frontend Overview Continuation.

(Fig. 5.5 - 1) Apdex Summary Pie Chart: This tile visually categorises user satisfaction levels across all web applications, dividing user experiences into distinct Apdex categories.


Search Dynatrace Demo Live: Demo Live...
Last 72 hours
6
178
User Experience Analytics EasyTravel
Data explorer
Data explorer
Query for metrics and transform results to gain desired insights. View Documentation
Pie
Advanced mode
A
Add Metric
Run query
Result
Save changes to dashboard
Pin to dashboard
Settings
Resolution
Show legend
Fold transformation
Action count (by Apdex c...
Unit
Format
Add color override

```
builtin:apps.web.actionCount.category:filter(and(in("dt.entity.application",entitySelector("type(application),entityName.contains(~"easytravel~")"
)))):splitBy("Apdex category")
```

You have unapplied changes. Rerun the query to apply them.

Timeframe: 2024-09-07 12:49 - 2024-09-10 12:49 3 record(s)

🟥 32.8k FRUSTRATED
🟩 2.44M SATISFIED
🟨 90.8k TOLERATING

Resolution: Auto
Fold transformation: Auto
Default
Unit: Auto
Format: Auto
SATISFIED 🟩
TOLERATING 🟨
FRUSTRATED 🟥

Figure 5.7: Data Explorer configuration for Web Applications tile

This tile is a prime example of one that requires more extensive configuration. In this case, it was initially built using the non-advanced mode, which allows you to select metrics in an intuitive way using drop-down menus. From there, I fine-tuned the query, toggling the advanced mode, as shown in the picture, making the necessary adjustments for the correct parsing of the entityName and the appropriate split. The split is then colorized using the settings menu on the right-hand side.

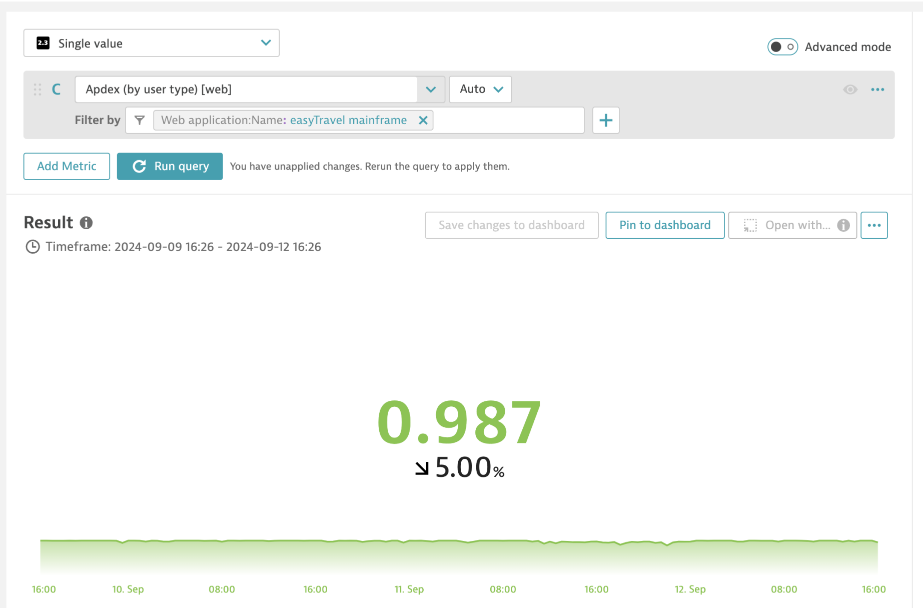
(Fig. 5.5 - 2) Satisfaction Line Chart: This tile tracks the evolution of user satisfaction over time.

```
builtin:apps.web.actionCount.category:filter(and(or(in(
"dt.entity.application",entitySelector("type(application),
entityName.contains(~"easyTravel~")")))))):splitBy
("Apdex category"):sort(value(auto,descending))
```

Listing 1: Data Explorer configuration for Web Applications tile

In this case, the query in Listing 1 is structured to pull user actions from the easyTravel infrastructure using the apps.web.actionCount.category metric. The query filters by the entity.application type and matches it with applications that contain “easyTravel” in their name. It then splits the data by Apdex categories—**Satisfied**, **Tolerating**, and **Frustrated**— and sorts the values in descending order.

(Fig. 5.5 & 5.6 - 3) Web Application Apdex Scores: Each web application within the infrastructure is represented by a tile showing its Apdex score. These tiles also contain direct links to a detailed analyses page within Dynatrace for further exploration, especially useful when the score indicates potential issues.


Single value

Advanced mode

C Apdex (by user type) [web] Auto

Filter by Web application:Name: easyTravel mainframe

You have unapplied changes. Rerun the query to apply them.

Result

Timeframe: 2024-09-09 16:26 - 2024-09-12 16:26

0.987

↘5.00%

Figure 5.8: Data Explorer configuration for Web Applications tile

These tiles are able to be created by using the non-advanced mode in the data Explorer, as can been seen in Figure 5.8 (Upper-right corner toggle). By toggling the advanced mode, the drop downs transform into a text box with a query, as is demonstrated in Figure 5.7 and, from there on, as the query itself in code listings.

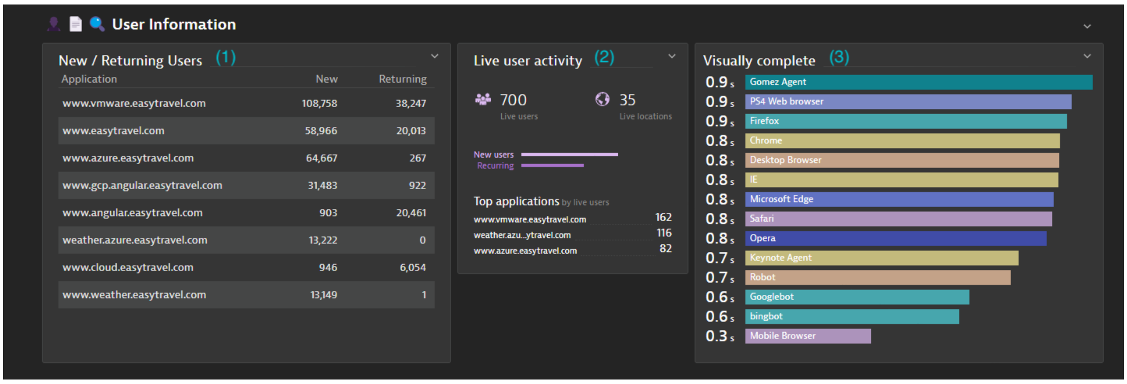
Questions Review

The **Frontend Overview** section above, only focuses on the first question.

How many users are satisfied according to the Apdex? Through the first pie chart in Dashboard in Fig. 5.5 (1) I can summarise that 2.34M users are satisfied throughout the easyTravel sites. The subsequent number charts focus on single applications instead, which allows me to easily narrow down where potential issues are.

User Information Section

The User Information Section showcases real-time engagement metrics, such as new versus **returning users**, live user activities, and performance metrics like **visually complete** times.



User Information

New / Returning Users (1)

Application	New	Returning
www.vmware.easytravel.com	108,758	38,247
www.easytravel.com	58,966	20,013
www.azure.easytravel.com	64,667	267
www.gcp.angular.easytravel.com	31,483	922
www.angular.easytravel.com	903	20,461
weather.azure.easytravel.com	13,222	0
www.cloud.easytravel.com	946	6,054
www.weather.easytravel.com	13,149	1

Live user activity (2)

700 Live users

35 Live locations

New users
Recurring

Top applications by live users

www.vmware.easytravel.com	162
weather.azu...ytravel.com	116
www.azure.easytravel.com	82

Visually complete (3)

Figure 5.9: Dashboard 1 User Info Section

(Fig. 5.9 - 1) New/Returning Users by Web Application: This tile breaks down how many new and returning users are interacting with each web application within easyTravel.

```
SELECT useraction.application AS "Application",
       condition(count(*), where newUser is true) as "New",
       condition(count(*), where newUser is false) as "Returning"
FROM usersession
WHERE userType = "REAL_USER"
  AND useraction.application LIKE "%easytravel%"
GROUP BY useraction.application
```

Listing 2: User Sessions Query for Returning / New Users

This tile is built using a USQ as shown in Listing 2, the conditions check whether a user is flagged as new or returning and counts them. I also filter out synthetic data by only selecting a real user type and refine it to any easyTravel applications.

(Fig. 5.9 - 2) Live User Activity: This built-in tile gives an overview of currently online users and their locations as well as a different way of displaying recurring users and the most viewed websites. It is configured through the tile menu within the dashboard.

(Fig. 5.9 - 3) Visually Complete Times: As shown in Listing 3, by excluding synthetic monitors we can illustrate how long it takes for content to become visually complete for real users, categorized by their browser. While categorizing by browser isn't strictly necessary, my experience with cross-browser compatibility suggests that longer visual completion times are often associated with specific browser-related issues. This approach also makes it easier to identify trends, rather than risking a focus on outliers.

```
builtin:apps.web.visuallyComplete.load.browser:filter
(and(or(in("dt.entity.browser",entitySelector("type(browser),
not(entityName.contains(~"Synthetic~"))")),or
(in("dt.entity.application",entitySelector
("type(application),entityName.contains
(~"easytravel.com~")"))))):splitBy("dt.entity.browser")
:sort(value(auto,descending)):limit(20)
```

Listing 3: Data Explorer configuration for Visually Complete times

Services and Error Levels Section

In this section, we're monitoring services and error rates. The goal is to make it visually simple for someone to quickly identify issues. For instance, in the attached image, if I were routinely checking the infrastructure, I could easily spot a service-wide uptick in requests around April 3rd. This would prompt me to investigate further, as the chance of potential issues being amplified is much higher. If issues were detected, one would either move to more specific problem management tools within Dynatrace or, in our team's case, pass the issue on to a specialized team—usually in-house for the client—that handles such problems. Our role is primarily focused on monitoring and generating regular reports so the client can address any identified issues.

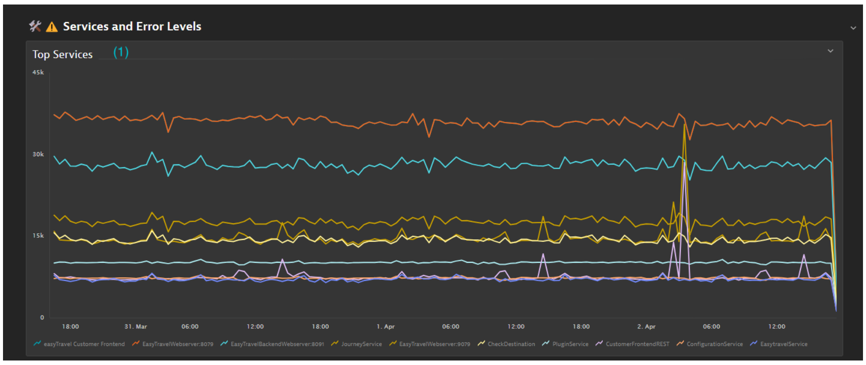
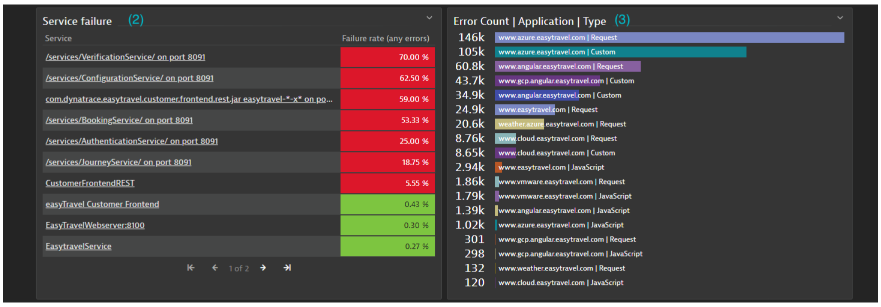


Figure 5.10: Dashboard 1 Services and Errors Section Part 1



Service failure (2)

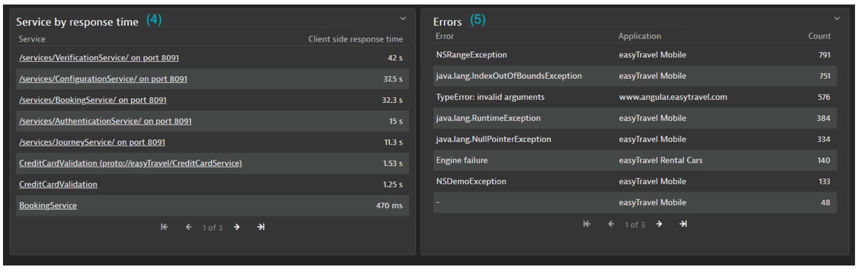
Service	Failure rate (any errors)
/services/VerificationService/ on port 8091	70.00 %
/services/ConfigurationService/ on port 8091	62.50 %
com.dynatrace.easytravel.customer.frontend.rest.jar easytravel-*-x* on po...	59.00 %
/services/BookingService/ on port 8091	53.33 %
/services/AuthenticationService/ on port 8091	25.00 %
/services/JourneyService/ on port 8091	18.75 %
CustomerFrontendREST	5.55 %
easyTravel Customer Frontend	0.43 %
EasyTravelWebserver:8100	0.30 %
EasytravelService	0.27 %

1 of 2

Error Count | Application | Type (3)

146k	www.azure.easytravel.com | Request
105k	www.azure.easytravel.com | Custom
60.8k	www.angular.easytravel.com | Request
43.7k	www.gcp.angular.easytravel.com | Custom
34.9k	www.angular.easytravel.com | Custom
24.9k	www.easytravel.com | Request
20.6k	weather.azure.easytravel.com | Request
8.76k	www.cloud.easytravel.com | Request
8.65k	www.cloud.easytravel.com | Custom
2.94k	www.easytravel.com | JavaScript
1.86k	www.vmware.easytravel.com | Request
1.79k	www.vmware.easytravel.com | JavaScript
1.39k	www.angular.easytravel.com | JavaScript
1.02k	www.azure.easytravel.com | JavaScript
301	www.gcp.angular.easytravel.com | Request
298	www.gcp.angular.easytravel.com | JavaScript
132	www.weather.easytravel.com | Request
120	www.cloud.easytravel.com | JavaScript

Figure 5.11: Dashboard 1 Services and Errors Section Part 2



Service by response time (4)

Service	Client side response time
/services/VerificationService/ on port 8091	42 s
/services/ConfigurationService/ on port 8091	37.5 s
/services/BookingService/ on port 8091	32.3 s
/services/AuthenticationService/ on port 8091	15 s
/services/JourneyService/ on port 8091	11.3 s
CreditCardValidation (proto://easyTravel/CreditCardService)	1.53 s
CreditCardValidation	1.25 s
BookingService	470 ms

1 of 3

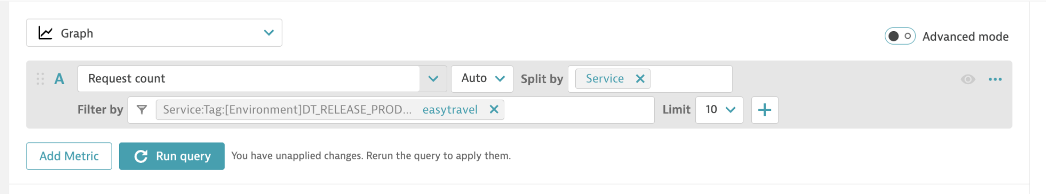
Errors (5)

Error	Application	Count
NSRangeException	easyTravel Mobile	791
java.lang.IndexOutOfBoundsException	easyTravel Mobile	751
TypeError: invalid arguments	www.angular.easytravel.com	576
java.lang.RuntimeException	easyTravel Mobile	384
java.lang.NullPointerException	easyTravel Mobile	334
Engine failure	easyTravel Rental Cars	140
NSDemoException	easyTravel Mobile	133
-	easyTravel Mobile	48

1 of 3

Figure 5.12: Dashboard 1 Services and Errors Section Part 3

(Fig. 5.10 - 1) Top Services Multi-Line Chart: Shows the top services by request over a given time-frame, each application is represented by a line. Illustrated in Fig. 5.13.



Graph

Advanced mode

A Request count Auto Split by Service

Filter by Service:Tag:[Environment]DT_RELEASE_PROD... easytravel Limit 10

You have unapplied changes. Rerun the query to apply them.

Figure 5.13: Data Explorer configuration for the Top Services Chart

(Fig. 5.11 - 2 & 3) Services Failure Rate % and (Fig. 5.12 - 4 & 5) Top Error List: This tile shows top failing services, the error count per application and the type of request it originates from, the services taking the longest to respond and the most frequent errors.

The service failure rate and **services by response time** are configured through the usual drop down menu selections;

The Top Errors List is configured as a User Sessions Query as shown in Listing 4:

```
SELECT name AS Error, application AS Application,
COUNT(*) AS "Count"
FROM usererror
WHERE application LIKE "%easyTravel%"
OR application LIKE "%easytravel%"
GROUP BY Error, Application
```

Listing 4: User Sessions Query for service failure rate and response time

The count of errors by the application they appear in is done through a relatively simple USQ as shown in Listing 5;

(Fig. 5.11 - 3) Error Count | Application | Type

```
builtin:apps.web.countOfErrors:filter(and(or(in
("dt.entity.application",entitySelector("type(application),
entityName.contains(~\"easytravel.com~\")")))))
:splitBy("dt.entity.application", "Error type")
:sort(value(auto, descending))
```

Listing 5: Data Explorer configuration for the Error Count tile

Similarly done to the previous queries, only differing significantly in the domain of selection.

User Funnel

This section visualizes the user's journey through the easyTravel platform, from opening the website to booking a journey. It allows us to identify **conversion rates** and pinpoint where users drop off. An example of a User Funnel is shown in Figure 5.14.

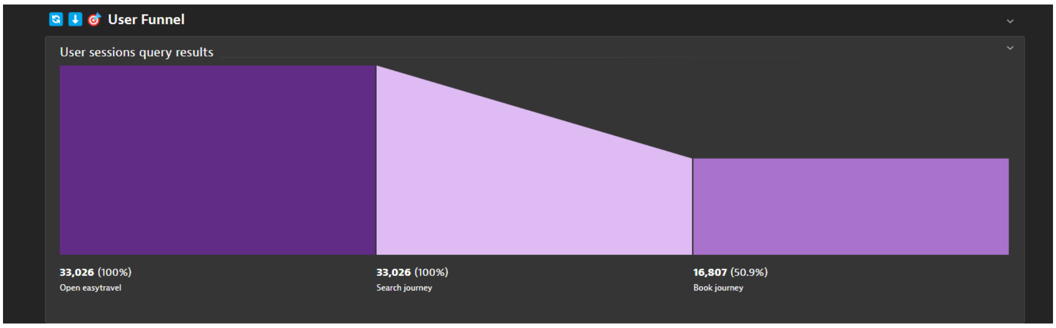


Figure 5.14: User Funnel for easyTravel

What is a User Funnel?

A User Funnel is a visualization that maps out the steps users take through a specific series of actions within an application, like purchasing a product or completing a registration. This funnel is created using a User Session Query (USQ) in Listing 6.

```
SELECT FUNNEL(
    useraction.name="AppStart (easyTravel)" AS "Open easytravel",
    useraction.name = "searchJourney" AS "Search journey",
    useraction.name = "bookJourney" AS "Book journey"
)
FROM usersession
```

Listing 6: User Sessions Query for the Funnel

What are conversion rates?

Based on my training, conversion rates are defined as the percentage of users who complete a desired action out of the total visitors. This action can vary—whether it's signing up for a newsletter, making a purchase, registering for an account, or any other key performance indicator (**KPI**) the client may be interested in.

Questions Review

We're able to answer almost all questions on this dashboard, leaving some for the secondary dashboard or, as I will determine at the end, unanswered.

How many users are satisfied according to the APDEX?
Dashboard in Figure 5.5, (1): 2.34M users are satisfied across the easyTravel applications.

Why and where did users exit the navigation?
Funnel in Figure 5.14: We observe a 49.1% drop-off at the third step of the funnel. This is quite common, especially based on personal experience, where many users "simulate" a booking before deciding whether to proceed with it. It's difficult to determine the exact reason for this drop-off without a more in-depth analysis. Referencing the tiles in **Figure 5.11, (2)(3) and Figure 5.12, (5)**, we can address questions such as "**which parts of the website encounter the most errors**" and "**is there a correlation between users who leave and those errors?**" From that information, it is clear that the number of total errors is insufficient to establish any meaningful correlation with user behavior, and the website with the most errors is unrelated to the booking page or process.

What activities do users perform the most on the website [...]?
Dashboard in Figure 5.10: The top services indicate that there are significantly more backend requests than frontend ones, and there was no easy way to distinguish between the two. However, the first relevant service on the list appears to be related to destination checking, which further supports the user funnel percentages.

How long does it take for them to be visually complete?
Dashboard in Figure 5.9, (3): Some modifications were subsequently approved by my manager to the question, allowing me to check which browsers take the longest to become visually complete across all services, rather than attempting to correlate them with the top services. This decision was made because, based on the team's experience, understanding which browsers take longer to load is ultimately more valuable to clients. The visually complete times are acceptable, coming in at just under a second across multiple browsers.

Are they returning users?
Dashboard in Figure 5.9, (1): Again slightly modified from the original question, I look at the total user breakdown across the top applications.

5.1.2 Secondary Dashboard

This dashboard focuses on the cloud.easytravel.com application, looking more into site-specific metrics, user behaviour and user origin.

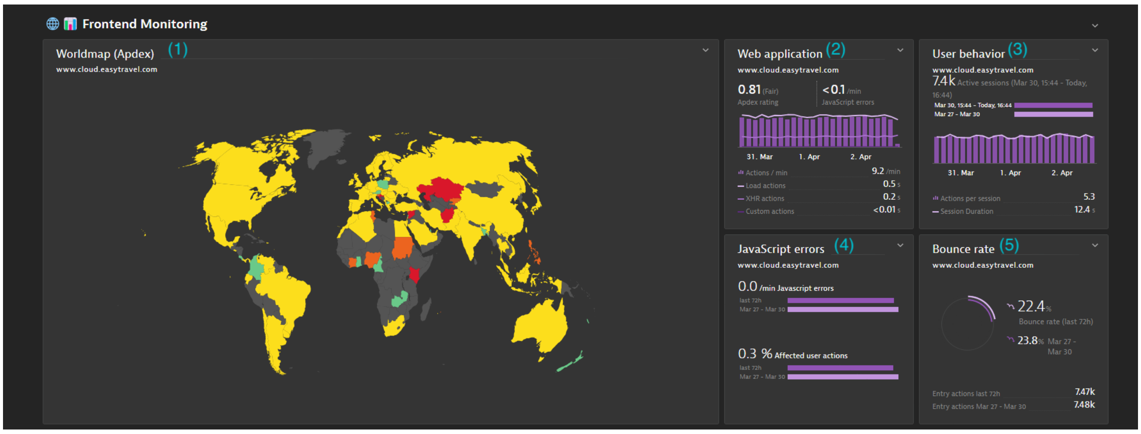


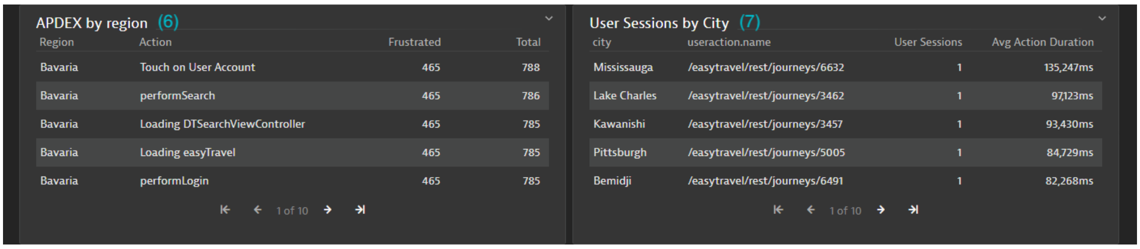
Figure 5.15: Dashboard 2 Frontend

(Fig. 5.15 - 1) World map by Apdex: A world map view of the various APDEX scores. Lower scores being limited to some regions might indicate that there is a localised issue.

- Green: The apdex score is excellent or good (≥ 0.85)
- Yellow: The apdex score is fair or poor (≥ 0.5)
- Red: The apdex score is unacceptable (≥ 0)

(Fig. 5.15 - 2 & 3) Web Application and User Behaviour: These tiles give us an overview of what's going on in real-time, the current APDEX score, the amount and type of actions, and how many user sessions we have active.

(Fig. 5.15 - 4 & 5) JavaScript Errors and Bounce rate: As one of our questions was in regards to there being a correlation between the errors and the bounce rate, these tiles show that such correlation is not possible in this case, as there is an insignificant amount of errors and the bounce rate is disproportionate to them.


APDEX by region (6)

Region	Action	Frustrated	Total
Bavaria	Touch on User Account	465	788
Bavaria	performSearch	465	786
Bavaria	Loading DTSearchViewController	465	785
Bavaria	Loading easyTravel	465	785
Bavaria	performLogin	465	785

User Sessions by City (7)

city	useraction.name	User Sessions	Avg Action Duration
Mississauga	/easytravel/rest/journeys/6632	1	135,247ms
Lake Charles	/easytravel/rest/journeys/3462	1	97,123ms
Kawanishi	/easytravel/rest/journeys/3457	1	93,430ms
Pittsburgh	/easytravel/rest/journeys/5005	1	84,729ms
Bemidji	/easytravel/rest/journeys/6491	1	82,268ms

Figure 5.16: Dashboard 2 Frontend Continuation

(Fig. 5.16 - 6) APDEX by region: This tile keeps track of the most frustrated regions around the world that are accessing the application. As a quick lookup, the total amount of user actions are also shown.

```
SELECT region AS "Region", useraction.name AS "Action",
    condition(count(*), where userExperienceScore = "FRUSTRATED")
    AS "Frustrated", count(*) AS "Total"
FROM session
WHERE region IS NOT NULL AND region NOT LIKE "%synthetic%"
GROUP BY Region, Action
ORDER BY Frustrated DESC
```

Listing 7: User Sessions Query for the APDEX by region

As shown in Listing 7, the tile is created using USQ and filters out synthetic regions as well.

(Fig. 5.16 - 7) User Sessions by City: This tile is more in line with the visually complete tiles from the main dashboard. It focuses on the actions taking the longest amount of time, where the user's action originates from and what the action path is.

```
SELECT city, useraction.name,
    count(*) AS "User Sessions",
    avg(useraction.duration) as "Avg Action Duration"
FROM session
WHERE city IS NOT NULL
GROUP BY city, useraction.name
ORDER BY avg(useraction.duration) DESC
```

Listing 8: User Sessions by City query

Similarly to above, as per Listing 8, we use a simple query and filter out the synthetic paths.

Questions Review

Answering the remaining questions can only be meaningful if done within single application, which is why this dashboard exists.

What is the geographical distribution of users?
Dashboard in Figure 5.15, (1): Through the world map we can determine the distribution of users, in relation to their APDEX scores.

The only question that remains unanswered is **How are users reaching the website?** After consulting with my team and manager, we concluded that the data required to answer this question cannot be retrieved based on my current training, and more experience is needed to address it. However, it is possible to obtain this information using other tools within Dynatrace, rather than relying solely on the dashboard, as they can sometimes be limiting, like in this case. This is also why the team maintains a bi-weekly report document with relevant information for some clients, rather than just relying on a dashboard.

5.2 HipsterShopLogs

As a contrast to the previous more varied environment, the HipsterShopLogs infrastructure consists in very few services and one web application. Due to this, the subdivision of the dashboards is not needed and all monitoring is done on the same dashboard.

The dashboard consequently only consists of:

a **Frontend Overview** section, focusing on the single web app's Apdex levels, as Figure 5.17;

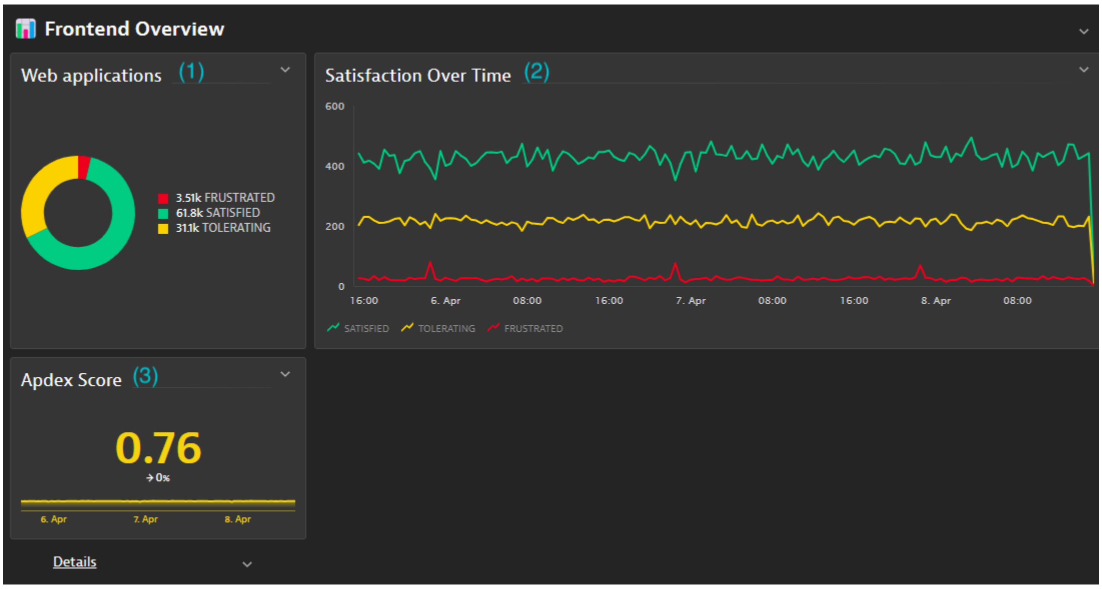
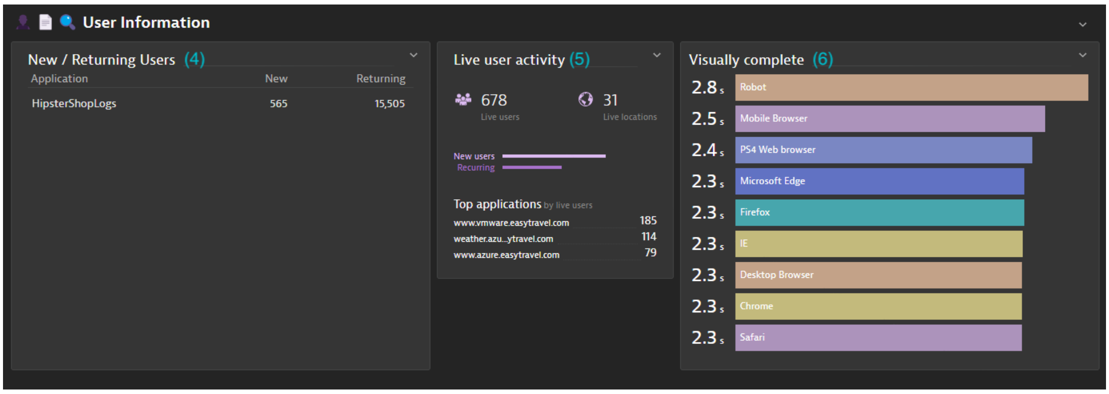


Figure 5.17: Dashboard 3 Frontend

a **User Information** section, showing new and returning users, the live user activity and the average time to be visually complete, as Figure 5.18;



Application	New	Returning
HipsterShopLogs	565	15,505

Top applications by live users

www.vmware.easytravel.com	185
weather.azu...ytravel.com	114
www.azure.easytravel.com	79

Figure 5.18: Dashboard 3 User Information

a **Services and Error Levels** section, showing the top services by request count, the service failure percentages, the error counts and a list of services by response time, as Figure 5.19 and Figure 5.20;

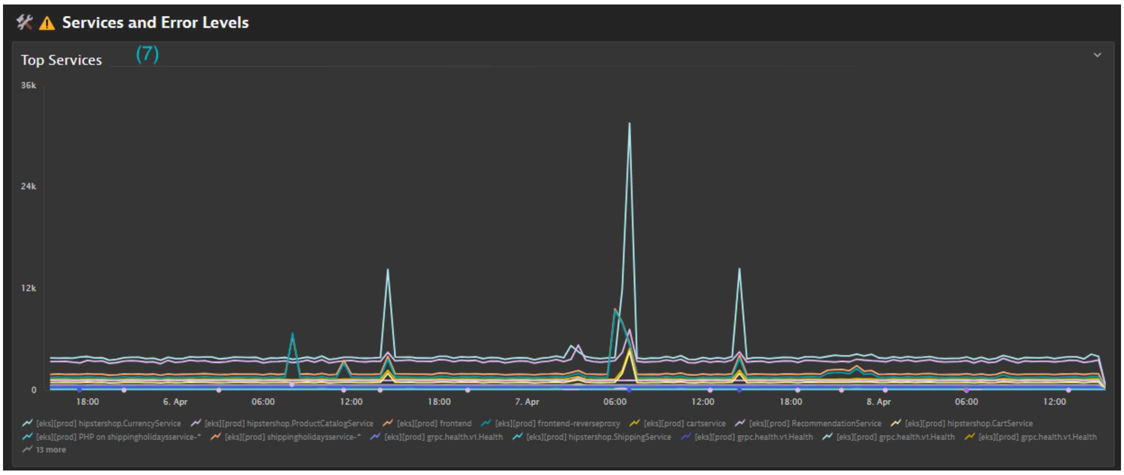
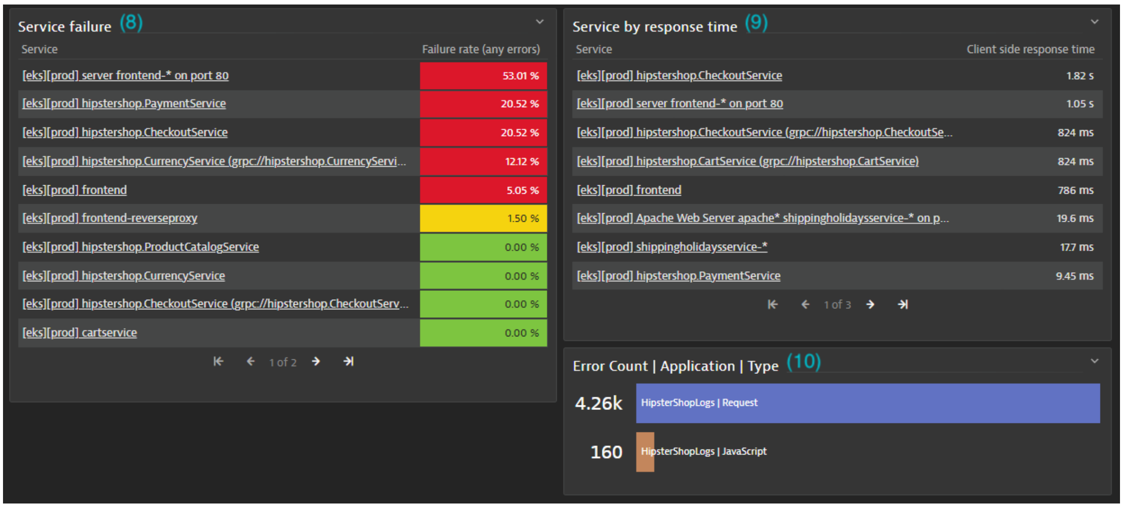


Figure 5.19: Dashboard 3 Services and Errors



Service failure (8)

Service	Failure rate (any errors)
[eks][prod] server frontend-* on port 80	53.01 %
[eks][prod] hipstershop.PaymentService	20.52 %
[eks][prod] hipstershop.CheckoutService	20.52 %
[eks][prod] hipstershop.CurrencyService (grpc://hipstershop.CurrencyServi...	12.12 %
[eks][prod] frontend	5.05 %
[eks][prod] frontend-reverseproxy	1.50 %
[eks][prod] hipstershop.ProductCatalogService	0.00 %
[eks][prod] hipstershop.CurrencyService	0.00 %
[eks][prod] hipstershop.CheckoutService (grpc://hipstershop.CheckoutServ...	0.00 %
[eks][prod] cartservice	0.00 %

1 of 2

Service by response time (9)

Service	Client side response time
[eks][prod] hipstershop.CheckoutService	1.82 s
[eks][prod] server frontend-* on port 80	1.05 s
[eks][prod] hipstershop.CheckoutService (grpc://hipstershop.CheckoutSe...	824 ms
[eks][prod] hipstershop.CartService (grpc://hipstershop.CartService)	824 ms
[eks][prod] frontend	786 ms
[eks][prod] Apache Web Server apache* shippingholidaysservice-* on p...	19.6 ms
[eks][prod] shippingholidaysservice-*	17.7 ms
[eks][prod] hipstershop.PaymentService	9.45 ms

1 of 3

Error Count | Application | Type (10)

4.26k	HipsterShopLogs | Request
160	HipsterShopLogs | JavaScript

Figure 5.20: Dashboard 3 Services and Errors Continuation

and a **User Funnel** going from the product catalogue, to the product page, to the cart and to a completed order page, as Figure 5.21.

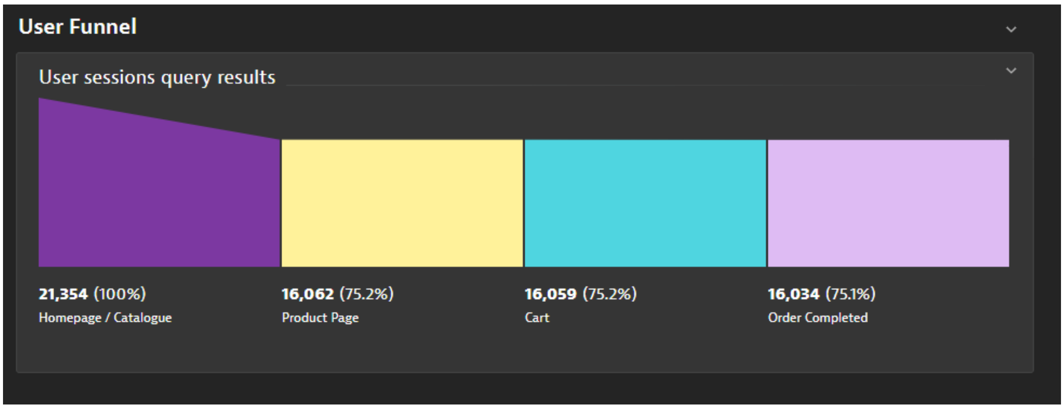


Figure 5.21: Dashboard 3 User Funnel

Questions Review

After consulting with my manager, I was advised to recreate the secondary dashboard based off of both previous ones. I was informed that this approach remains relevant to clients and, would usually be accompanied by a written report to address any additional questions the client may have.

How many users are satisfied according to the APDEX?
Dashboard in Figure 5.17, (1): There are 61.8K satisfied users.

Where did users exit the navigation?
User Funnel in Figure 5.21: There is a reasonable drop-off between people browsing the catalogue to actually viewing a product. Once they open a product they almost always add it to their cart and complete the order.

Which parts of the website error out the most?
Dashboard in Figure 5.20, (8): From the data, it is clear that the server frontend on port 80 is the part of the website experiencing the most errors, with a failure rate of 53.01%. This is followed by the HipsterShop PaymentService and HipsterShop CheckoutService, both at 20.52%. Other services like the CurrencyService and the general frontend are also encountering some errors, but at lower rates of 12.12% and 5.05%, respectively. This matches what we can determine by looking at the Service by response time (9) and the error count (10) tiles.

Is there a correlation between users that leave and those errors?
Dashboard in Figure 5.20, (8)(9): Although we have quite significant errors in important parts of the website, users seem to successfully complete their ordering process. There is no significant drop-off throughout the process. (User Funnel)

What are the top services?
Dashboard in Figure 5.19: The currency service receives the most request, possibly due to users with different currency settings browsing the website.

How long do users take to be visually complete on the application?
Dashboard in Figure 5.18, (6): Visually complete times are high across the

board, indicating that there are clear optimizations to be made and errors that need to be resolved.

What is the breakdown of new and returning users? Dashboard in Figure 5.18, (4): Only 565 users are new and 15505 are returning.

Chapter 6

Conclusions

During my internship at Alten, I had the opportunity to explore Application Performance Monitoring (APM) using Dynatrace, with a particular focus on Real User Monitoring (RUM) and how the APDEX score reflects user satisfaction. My main task involved creating custom dashboards that translated user interaction data and performance metrics into visual insights. These dashboards were designed to help clients, regardless of their technical expertise, monitor application performance and user experience in real time.

This experience helped me better understand how Real User Monitoring fits within APM and why it's important for making sure applications meet user expectations. I learned how metrics like the APDEX score can point out areas that need improvement and how monitoring setups can be adapted to suit different clients. While I didn't pick up many new technical skills, the internship helped me improve my understanding of user experience monitoring and how to present data in a way that's clear and useful for clients, essentially strengthening my business skills instead.

Overall, the internship gave me practical experience with Real User Monitoring and showed me how important it is to create monitoring solutions that meet each client's needs. It also highlighted how useful clear and simple data visualization is when it comes to tracking and improving user experience.

Bibliography

[1] Dynatrace, "What is apm?," June 2024.

[2] Gartner, "Application performance monitoring (apm)," 2024.

[3] Apdex Alliance, "Apdex history," 2024.

[4] N. Pollock and G. Campagnolo, *Subitizing Practices and Market Decisions: The Role of Simple Graphs In Business Valuations.* Oxford University Press, August 13 2015.

[5] GmbH, finanzen.net, "G | s&p 500 stocks | s&p 500 companies | s&p 500 values." Markets Insider, 2018. Retrieved June 28, 2018.

[6] Gartner, "Magic quadrants and marketscopes: How gartner evaluates vendors within a market." Gartner.com, February 2008. Archived from the original on March 12, 2006, Retrieved July 27, 2009.

[7] Gartner, "Gartner magic quadrant: Frequently asked questions," 2019. Accessed June 6, 2020.

[8] Gartner, "How markets and vendors are evaluated in gartner magic quadrants." Gartner.com, August 2019. Retrieved August 15, 2019.

[9] R. Kramer, "Research note: Dynatrace elevates data observability and monitoring with ai automation." Moor Insights & Strategy, January 22 2024. Retrieved 2024-01-22.

[10] M. Costello, "What's changed: 2023 magic quadrant for apm and observability." Best Network Monitoring Vendors, Software, Tools and Performance Solutions, July 31 2023. Retrieved 2023-10-04.

[11] R. DeFrancesco, "Dynatrace helps cloud software run at peak performance," *Forbes*, 2020. Retrieved 2020-07-24.

[12] diginomica.com, "In pursuit of general intelligence - dynatrace and the death of the dashboard." diginomica.com, March 29 2022. Retrieved 2022-07-16.

[13] Container Journal, "Dynatrace deepens ai ties to kubernetes," *Container Journal*, February 5 2020. Retrieved 2020-09-14.

[14] A. Alliance, "Apdex technical specification version 1.1," 2020. Accessed: 2024-10-01.

[15] P. Sevcik, “Defining apdex,” 2005. Retrieved from Apdex.org.

[16] Tech Seeking Human, “Disrupt or be disrupted, the story behind the dynatrace reinvention - bernd greifeneder,” 2020. YouTube video, featuring Bernd Greifeneder.

[17] Dynatrace, “What is causal ai? why this deterministic ai approach is critical to business success,” 2023. Accessed: 2024-10-13.

[18] The New Stack, “Why dynatrace says ai is the answer for ’single panel’ observability for cloud native,” February 13 2020. Retrieved 2020-09-14.

[19] T. R. Weiss, “New software intelligence apps unveiled by dynatrace,” February 21 2023.

[20] Dynatrace, “Digital experience monitoring,” 2024. Accessed: 2024-10-14.

[21] P. A. Networks, “What is digital experience monitoring?,” 2024. Accessed: 2024-10-14.

[22] Dynatrace, “Digital experience monitoring with dynatrace,” 2024. Accessed: 2024-10-14.

[23] Dynatrace, “Types of synthetic monitors,” 2024. Accessed: 2024-10-01.

[24] Dynatrace, “Dashboards classic,” 2024. Accessed: 2024-09-11.

[25] Dynatrace, “Upgrade to dashboards: Classic to latest,” 2024. Accessed: 2024-10-01.

[26] D. Community, “Data usql no-live vs data explorer with the same timeframe don’t match,” 2024. Accessed: 2024-10-01.

List of Figures

List of Listings