System Requirements

Jonathan Ahlström Ossian Gewert Jacob Jönsson Simon Persson
André Roxhage Felix Sundholm

March 6, 2025

ETSN15 Requirements Engineering

Group Gamma - EasyTrip

Contents

1 Introduction

1.1 Naming convention

The naming convention used for requirements are descriptive camelCase names for Functional, Quality and Data requirements.

1.2 Context Diagram

The context diagram (Figure 1) illustrates the systems key stakeholders in the inner domain. This includes travellers, travel agencies, airlines, IT maintenance and ad providers. In the outer domain, the context diagram shows marketing platforms, Google Maps for displaying maps and advertisers.

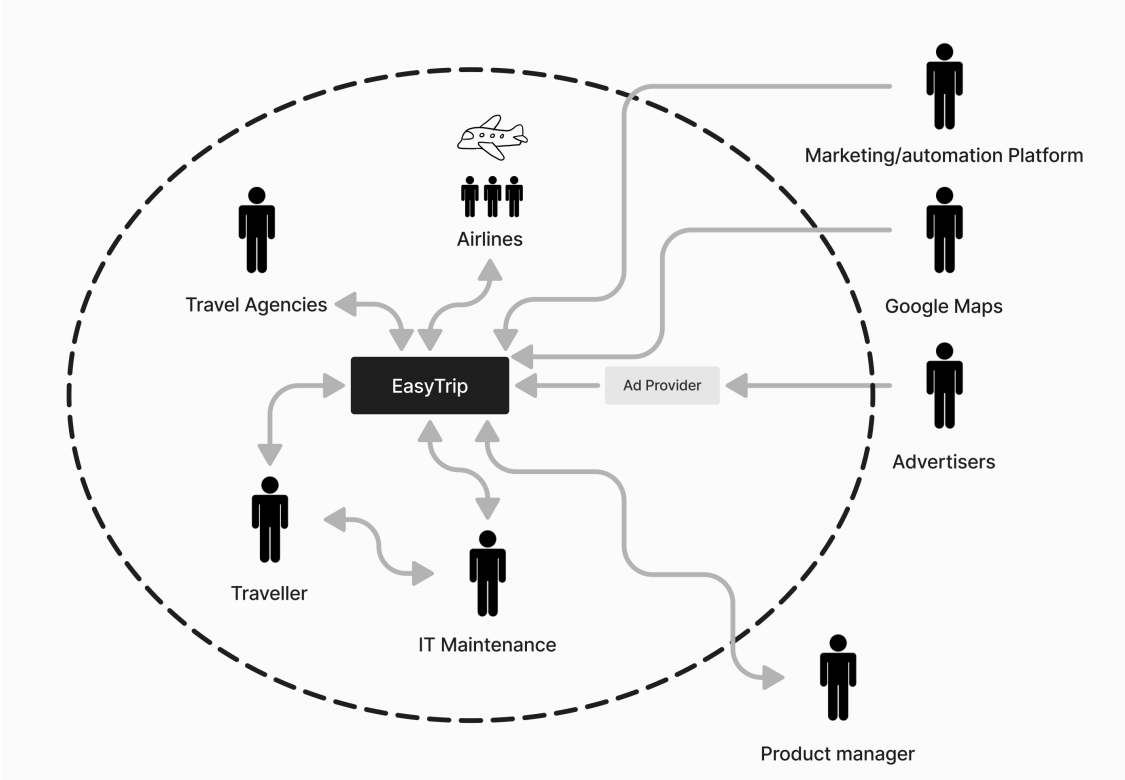

Figure 1: Context diagram

1.3 Business Goals

The system should claim 20% of the market share in the flight search market within the next 12 months. This will be done by differentiating the product from competitors and using unique data visualization. The system will gain a competitive advantage and attract customers from competitors and encourage them to fly more. The adaption of the product can be measured by tracking the number of new users in the system. See Figure 3 and Figure 4 for a visual representation of the system.

The system should include features that improve customer satisfaction and loyalty, leading to 30% of users coming back to visit the service at least once a month. The customer retention rates can be measured by user feedback and monitoring the flight search frequency by users.

The system should support features that enable the business to reach new markets and customer segments. This includes users from a wide range of economic status, from those seeking budget-friendly flights to those interested in premium travel options. This will be done by providing unique visualization of flight prices that attracts more customers.

2 Elicitation

2.1 Interviews of Travellers

In order to determine traveller goals and needs as well as to validate the product idea of Easytrip, data has been elicited from potential travellers. The data was collected from semistructured interviews with 2 participants. The participants were chosen based on people that we know are frequent travelers. The interviews were conducted in a casual setting, and the recorded data is anonymous.

An interview guide was used to ensure consistency among the interviews. The guide was structured in three parts: Introduction, experience with existing travel planning tools, questions regarding the process to finding cheap flight, as well as closing questions.

2.1.1 Results

The participants (N=2) had an average age of 24 years and both had similar prior experience with using travel planning tools. They both used airplane travel one or two times each ear, mostly for vacation. For shorter destinations, travel by car or train was desired to minimize their environmental footprint. Both believe that the most important factor when booking a flight is price, but other factors such as layovers and travel dates were also mentioned. Additionally, they value flexibility and personalization in the booking process.

2.1.2 Problems

One of the participants found it stressful how algorithms increase prices when refreshing the page. Booking through third-party platforms was also brought up as a problem since the particapnt had experienced multiple crashes. The other participant expressed frustration with the lack of transparency in pricing, with some sites having hidden baggage fees and unclear total costs making price comparisons between flights difficult.

2.1.3 Finding Cheap Flights

Participants used different methods to find cheaper flights. One preferred sorting by the lowest available prices and using price prediction tools like those in Hopper. The other valued flexible date search options and price calendars that displayed price variation over time. This suggests a need for a tool that combines multiple features to find the best deals hinting for a flexible design depending on user preferences.

2.2 Stakeholder Analysis

Our stakeholders include competitors such as Momondo and Flight Scanner, travel agencies providing data partnerships, airline companies indirectly benefiting from bookings, end travellers, travelers, providing feedback, IT maintenance and development teams, and non-obvious actors such as the support department. Each stakeholder have different needs and goals, and it is important to understand these in order to create a successful product as well as to manage the relationships and expectations of the stakeholders.

2.2.1 Airlines

Since airlines are EasyTrips main source of data, information is needed regarding API endpoints for fetching price, date and times, availability based on travellers' search queries. Also, agreements on business models for affiliate links will be needed to ensure revenue channels. Additionally, real-time updates on flight status and any changes in schedules will be required to maintain accurate and up-to-date information for our travellers.

Airlines benefit from increased visibility, effectively using EasyTrip as a marketing and sales channel. By providing accurate and real-time data, airlines can attract more customers and increase

their revenue through the platform. The competition between airlines is also reflected in the platform, as travellers will compare prices and services, increasing the pressure on airlines to provide competitive offers and maintain high service quality.

The platform has a strong dependency on airlines for providing accurate and timely data. Inconsistent or outdated flight information could negatively impact traveller experience and trust in our platform. Changes in airlines' API or business models could lead to disruptions and additional development costs and maintainment costs on our end. Airlines could withdraw from agreements, affecting our affiliate revenue stream. Any inaccuracies or delays in data could reflect poorly on their services and reputation.

Airlines may suggest improvements such as dynamic pricing or marketing campaigns to maximize mutual benefits. Potential collaboration could be prioritizing a airline in the search results, or offering exclusive deals to travellers. These collaborations could be beneficial for both parties, as it would increase the visibility of the airline and create a new revenue stream for us.

2.2.2 Travel Agencies

Travel agencies provide data partnerships and EasyTrip needs access to their data regarding flight offers in a similar way that we are dependent on the airlines' data.

Travel agencies benefit by reaching more customers through the platform in a similar way as for airlines. The main difference is the advantage of being able to combine multiple airline in routes including multiple layoveers. This is a service that airlines do not provide. Therefor, travel agencies can promote deals and lead customers to additional purchases since they also provide accommodations and other travel services. Both airline and travel agencies benefit from increased visibility and bookings through the platform, effectively using us as a marketing and sales channel.

There are risks that travel agencies could provide inconsistent data, leading to inaccurate package pricing or availability issues. Competition between agencies could result in disputes over featured listings. Also, there is a risk of agencies withdrawing from agreements, affecting our affiliate revenue stream. The platform is dependent on the travel agencies for providing accurate and timely data. Similar to the airline can changes to API or business models could lead to disruptions and additional development costs and maintainment costs on our end. Favoring of competitors could also be a risk, as the agencies might prioritize other platforms over EasyTrip. The agencies might have exclusive deals and partnerships with other platforms, which could limit the provided data.

Similar to the airlines is the potential to include promotional campaigns targeting high-demand destinations. This could be beneficial for both parties. However, such agreement would need to be carefully negotiated to ensure that both parties benefit from the collaboration, without hurting the relationship to other stakeholders. For the development process, travel agencies could also provide feedback on the platform and suggest improvements to the traveller interface or features to ensure that their data is presented in the best possible way.

2.2.3 Travellers

To provide a personalized experience, the platform needs information about travellers budget constraints, destinations and travel dates. Optionally, personal data such as email can be gathered to enable notifications on price changes. Also, feedback on their traveller experience crucial for the improvement and future of EasyTrip.

Travellers benefit from accurate, real-time price comparisons, and personalized recommendations. They gain time efficiency and cost savings when planning trips. Additionally, travellers are the main source of revenue for the platform, as they generate bookings and affiliate revenue. Therefore, user satisfaction and loyalty are crucial for the success of the platform. By providing feedback and suggestions, travellers can influence the platform's development and improve their overall experience. This could lead to increased retention and engagement, meaning travellers will return to the platform for future travel planning.

The dependency on airlines is a factor that can generate issues such as inaccurate flight information, delayed notifications, or privacy concerns. If travellers feel their data is not secure or their preferences are not being met, they may lose trust in the platform and seek alternatives.

To mitigate these risks, EasyTrip will implement validation and consistency checks to ensure accurate flight information. Additionally, the platform will provide clear and transparent privacy policies to address any privacy concerns. Regular traveller feedback will also be gathered by interviews and forums to identify possible problems with the system.

2.2.4 IT Support and Maintenance

The IT support team requires a comprehensive administration interface to monitor system errors, API updates, and data quality from airlines and travel agencies. They need access to logs, error reports, and real-time data feeds to ensure the system's integrity and performance. Since payment and booking are handled by airlines or travel agencies, the IT support team does not need access to payment information, allowing them to focus on technical issues and system maintenance.

The IT support team is crucial for maintaining the platform's reliability and performance. They ensure that data from airlines and travel agencies is accurate and up-to-date, which is essential for traveller trust and satisfaction. By resolving technical issues promptly and ensuring seamless API integrations, the IT support team helps maintain the platform's reputation and operational efficiency.

The IT support team faces risks such as system downtime, and technical issues that could disrupt the platform's functionality. Not enough monitoring and delayed response to issues can lead to prolonged outages, negatively impacting traveller experience and trust. Additionally, the team must stay updated with API changes from airlines and travel agencies to prevent integration issues. To resolve this, a error handling system must be in place to detect and address issues promptly.

To mitigate these risks, the IT support team will implement robust monitoring and alerting systems to detect and address issues promptly. Regular updates and maintenance schedules will be established to ensure system stability. Collaboration with airlines and travel agencies will be maintained to stay informed about API changes and updates. Additionally, the team will conduct regular data quality checks and validation processes to ensure the accuracy and reliability of the information presented on the platform.

2.2.5 Competitors - Momondo, Flight Scanner

EasyTrip needs detailed information about the features, pricing models, and user interfaces of competitors like Momondo and Flight Scanner. Specifically, insights into how they handle flight data aggregation, display recommendations, and optimize search algorithms are critical. Also, understanding their their business model will be important since EasyTrip will have similar revenue streams as the competitors, using affiliate marketing strategies and partnerships with airlines.

Competitors may benefit from increased visibility and bookings, similar to EasyTrip. However, their primary stake is competitive, as they aim to attract and retain a larger user base. Their ability to secure exclusive partnerships may impact our access to data and revenue streams. However, since EasyTrip is rather for finder wherer the cheapest flight is, the competition has a different perspective on the market. They are more focused on providing a flight price comparisons for travelers already knowing their destination.

Competitors have several risks to EasyTrip, such as attracting potential travellers with more competitive pricing, better deals, or superior user experience. Established competitors can also secure exclusive data partnerships, limiting EasyTrip's access to certain deals or flight information. Also, their already established brand loyalty can make it challenging for EasyTrip to gain market shares.

To minimize the risks, EasyTrip can focus on differentiating itself through innovative features, such as visualization of flight data. EasyTrip can also emphasize user experience by providing intuitive search tools. Collaboration with smaller travel agencies or airlines could help secure

unique offers that competitors may not have access to. EasyTrip can also leverage feedback from users to continuously improve its platform and stay competitive.

2.3 Conflicting Stakeholder Interests

The table below explains the key conflicts between stakeholders and possible solutions for resolving them:

Stakeholders Involved	Conflicting Interests	Potential Solutions
Airlines vs. Travel Agencies	Airlines prefer direct bookings, while travel agencies benefit from combining routes	Negotiate fair partnerships where travel agencies promote airline offers and bundle other services effectively.
Airlines vs. EasyTrip	Airlines may want preferential treatment in search results	Maintain neutrality by implementing a transparent ranking system based on user preferences, not payments.
Competitors vs. EasyTrip	Competing for user base and exclusive airline partnerships	Differentiate through unique features, personalized services, and feedback-driven innovation.
Travellers vs. Airlines	Travellers want low prices, while airlines focus on maximizing profit	Implement dynamic pricing with filters to show budget-friendly options alongside premium offers.
IT Support vs. Airlines/Agencies	API changes by airlines or agencies may cause disruptions	Maintain regular communication with API providers and implement fallback mechanisms to reduce downtime.

2.3.1 Resolving Conflicts

To resolve conflicts, EasyTrip will establish internal clear and transparent guidelines for collaboration and prioritize creating beneficial agreements for multiple stakeholders. Keeping strong relationships using regular communication with airlines and travel agencies is important, especially due to our dependencies on their data. Feedback from travellers will probably be the greatest prioritization, ensuring user satisfaction without compromising data partnerships. A robust monitoring system will be implemented to manage technical dependencies and reduce the risk of conflicts affecting platform performance.

2.3.2 Alternative Data Sources - Flight Scanner API

In case of losing access to airline data, EasyTrip will use Flight Scanner API as an alternative data source. Flight Scanner provides similar data on flight prices, availability, and schedules, which could be used as a backup plan. However, this would require a different business model, as Flight Scanner does not provide affiliate marketing strategies. Instead, EasyTrip could rely solely on ads as a revenue stream. This alternative data source could be used as a backup plan in case of disruptions, changes in airline data access and for distribution of risk.

2.4 Elicitation of easyJet

To further enhance our stakeholder analysis and gather more information about the stakeholders, both competitors such as Momondo and Flight Scanner, as well as airlines such as SAS and Norwegian and 45 other airlines, were contacted. The purpose of the contact was to gather information about their needs, pricing strategies, and API endpoints.

EasyJet is a low-cost airline that operates in Europe. The airline was contacted via email and questions were asks about traveler challenges, API endpoints and limitations, data display requirements and other considerations. The following information was gathered from the contact:

2.4.1 Challenges Faced by Travellers

EasyJet highlighted that while its booking system is designed to be user-friendly, travelers often face difficulties when trying to manage bookings across different platforms such as third party solutions like Travel Agencies. Seamless integration of third-party platforms with airline booking systems is a key challenge that can be improved upon to enhance the user experience.

2.4.2 API Integration and Real-Time Data

EasyJet pointed out the importance of API integration that supports real-time updates on flight schedules, prices, and availability. While they did not reveal specific technical details about their API agreements, they recommended that any travel planning tool prioritize accuracy, data privacy, and security. This ensures users have up-to-date information and maintain control over their personal data.

2.4.3 Display Requirements

Although easyJet did not provide detailed requirements for how their flights and promotions should be displayed on third-party websites, they underscored the importance of maintaining high data standards and compliance. Platforms like EasyTrip would need to adhere to data usage agreements and provide transparent privacy policies.

2.4.4 Use of Publicly Available Resources

Due to prioritization on enhancing user experiences on their own platforms, easyJet was unable to offer a dedicated meeting. Also they were unable to share details about their pricing strategies or data-sharing agreements with our competitors. However, they encouraged us to use publicly available resources on their website for information regarding booking processes, and user services.

2.4.5 Underutilized Features in Flight Comparison Tools

Although easyJet did not specify which features they believe are underutilized in current flight comparison systems, they truly emphasized on data accuracy, privacy, and ease of booking management. This suggests that addressing these areas could lead to significant improvements.

2.4.6 Takeaways

1. EasyTrip will need to explore API partnerships that prioritize real-time data updates, potentially collaborating with services like Flight Scanner as a backup plan or as the primary data source in the minimal viable product.
2. Enhancing seamless management of bookings across platforms could address key pain points identified by easyJet. Excluding booking features could be a strategic decision to avoid direct competition with airlines and to focus on providing a user-friendly flight comparison tool. This resolve the conflict of interest between airlines and EasyTrip.
3. Implementing clear data privacy policies and ensuring compliance with third-party agreements will be critical to avoid any data-related issues. This will help build trust with airlines and travelers, and maintain a positive reputation in the industry. The use of cookies and data collection should be transparent and in accordance with GDPR regulations to favor the travellers.

3 Requirements Specification

In the following subsections, we will go through each functional requirement and the quality requirements associated with it. The main focus will be to achieve a minimal viable product that satisfies the functional requirements while meeting the quality requirements. The reason for this is that we want to be able to release the product as soon as possible and then iterate on it. Therefor we strive to balance technical feasibility, user experience, security, and other quality characteristics.

The requirements in this document follow a hierarchical structure where indented requirements represent data or quality requirements that are connected to their parent functional requirement. This structure helps show the relationships between functional requirements and their associated quality attributes or data specifications.

For example, a functional requirement like **HeatmapSearch** may have indented quality requirements such as **HeatmapPerformance** that specify performance criteria for that feature.

3.1 Heatmap Visualization

HeatmapSearch: The system shall display a heatmap that visually represents flight prices for various destinations, with the lowest available prices highlighted using colors, engaging users in the search process. See an example in Figure 4. This can be described as a task:

Task: Find cheap trip destination using heatmap
Subtasks: 1. Input origin / allow location data 2. Input flight dates 3. Navigate heatmap 4. Select area 5. Select show details for flight 6. Get referred to airline
Variants: 1a. User types in origin airport 1b. User uses device location data for origin

Task description 1: Finding a cheap trip destination using the heatmap

> **HeatmapPerformance**: The heatmap should update dynamically with good performance after data input or changes such as price range. For updates that require a new response from the server, the server should send the response within 1.5 seconds of receiving the query. For updates that are done locally on the client, the change should be done within 3 seconds on the reference testing devices.
>
> Monitor UI logs to confirm rendering completion time after changes to input data. Measure server response times with server-side logging.
>
> **HeatmapVisibility**: The heatmap shall be responsive and maintain readability on screen sizes ranging from 320px (mobile) to 1920px (desktop). It shall use a color gradient that ensures a minimum contrast ratio of 4.5:1 (WCAG AA) for text labels and distinguishable color differences between price categories. See Figure 3 and Figure 4. Note the examples are only prototypes and are not final.

DateRangeMap: Travellers and Travel Agencies can change the dates for which the map shows data. Domain-level control for flight search customization.

> **DateChangeResults**: The map should reflect date changes with good performance without requiring a full reload. For updates that require a new response from the server, the server should send the response within 1.5 seconds of receiving the query. For updates that are done locally on the client, the change should be done within 2 seconds on the reference testing devices.

Monitor UI logs to confirm rendering completion time after date modifications. Measure server response times with server-side logging.

FlexibleRangeMap: The system should support flexible date ranges spanning at least 30 days for price comparisons.

Test by selecting a range of 30-60 days, verifying the system returns results across all days without errors.

ListFlightsFromMap: Travellers and Travel Agencies can see a text list of destinations with the lowest flight prices shown on map. Product-level display of destinations based on search.

TextListsShown:Destination lists should be updated in real-time when new flight deals become available.

Insert a new deal in the back-end or get deal from airlines API and verify update propagation within milliseconds.

AirportSelection: Travellers and Travel Agencies can select airports from the heatmap. Product-level feature allowing interactive selection.

FastAirportSelection: Selected airports should be highlighted within 0.4 seconds on the reference testing devices and updated search results should be sent by the server within 1.5 seconds of the query being received.

Measure UI update times via browser dev tools or performance logs. Measure server response times with server-side logging.

3.2 User Security

LawCompliance:The system should comply with general data protection regulations (GDPR). Domain-level requirement ensuring user data privacy.

GDPRCompliance: Perform a data protection regulations (GDPR) audit to ensure compliance with data protection regulations.

Document and address any issues found during the audit.

SecureDataEncryption: User data should be encrypted both in transit and at rest. Domain-level security requirement.

DataEncryptionCheck: Ensure that all user data is encrypted using secure protocols and stored securely.

Perform security audits and penetration tests to verify encryption and storage mechanisms.

3.3 Flight Search

SearchCheapFlights: Travellers and Travel Agencies can find destinations with low flight prices when providing a departure city and dates. Search results shall be ranked by price in ascending order, ensuring the most cost-effective options are prominently displayed. Goal-level requirement defining a key system function. This can be described as a task:

Task: Find cheap flight
Purpose: Find cheap flight to general area
Input parameters: • Origin • Destination • Dates
Display options: • Heatmap • List • Filter • Date sliders

Task description 2: Finding a cheap flight

> **FastSearchResults**: Search results should be displayed with good performance for queries that include a departure city, and travel dates, retrieving data for up to _ available flights. For updates that require a new response from the server, the server should send the response within 1.5 seconds of receiving the query. For updates that are done locally on the client, the change should be done within 1.5 seconds on the reference testing devices.
>
> Tested by generating random inputs and measuring response time using automated performance testing tools, logging response times for 1000+ random queries.

ListCheapFlights: The system shall present a structured list of available flights to a specified destination, sorted by price in ascending order. Product-level feature for displaying search results.

> **TextListSupport**: The list should support sorting by price, duration, and airline, with sorting applied in under 1 second on the reference testing devices.
>
> Use UI performance tests to time sorting actions on datasets of varying sizes for example destinations with a lot of flights or destinations with less flights.

SearchCheapDestinations: Travellers and Travel Agencies can find flights to a given destination. Domain requirement focusing on the flight search feature.

> **FastPriceDisplay**: The system shall retrieve and display flight prices with good performance for search requests that include a departure city, and travel dates, processing up to _ flight options per request. For updates that require a new response from the server, the server should send the response within 1.5 seconds of receiving the query. For updates that are done locally on the client, the change should be done within 1.5 seconds on the reference testing devices.
>
> Execute automated tests simulating 1000+ searches and log the response time until results are fully rendered. Measure server response times with server-side logging. Primarily same as *FastSearchResults*.

DateRangeSearch: Travellers and Travel Agencies should be able to provide a range of possible dates for their flight search making the service more flexible and appealing to a wider range of users. Domain-level functionality supporting flexible searches, in flight search feature.

> **DateRangeCheck**: The system should support flexible date ranges spanning at least 30 days for price comparisons.
>
> Test by selecting a range of 30-60 days, verifying the system returns results across all days without errors.

GeolocationService: Travellers can have their device provide their current location. Domain-level feature leveraging geolocation services.

GeoLocationCheck:Location detection should be completed in under 3 seconds with at least 95% accuracy on the reference testing devices.

Use GPS data from 50 or more test cases across different devices and measure accuracy based on expected vs. detected locations.

ShowFlights: Traveller can click through to ticket booking page for shown flights. Product-level requirement enabling ticket purchase.

CorrectNavigation: Clicking a flight should navigate to an external booking page for the shown flight.

Test clicking on flights and verify that the booking page shows the expected flight.

SaveFavoriteCity:A signed-in traveller can save favorite cities. Product-level feature for personalization.

FavoriteCityFetching: Favorite cities should be stored in under 5 seconds and retrievable in under 1 second.

Measure web and database server response times with server-side logging.

3.4 User Experience

UsabilityScore: The ease of use of the product shall be tested and must achieve a System Usability Score (SUS) of at least 70 points, indicating acceptable usability.

SUSTesting: Conduct usability tests with a sample group of users and calculate the SUS score.

Ensure the average score meets or exceeds 70 points.

AccessibilityCompliance: The accessibility of the product shall be tested and must comply with WCAG 2.1 standards.

WCAGTesting: Perform accessibility audits using automated tools and manual testing to ensure compliance with WCAG 2.1 standards.

Document and address any issues found during testing.

RegularUserFeedback: The system shall regularly prompt users for feedback. Product-level feature that aims to gather user insights and improve customer satisfaction by understanding user experiences and expectations.

FeedbackCheck: After _ hours of use, the system shall prompt the user with the question "What do you think about EasyTrip?" and a rating scale.

3.5 Account Management & Personalization

AccountCreation: Travellers should be able to create accounts. Product-level requirement for account management.

AccountData - Data dictionary: Product-level requirement:
Class: User
The user is a Traveller or Travel Agency who has a user account in the product.

Examples: 1. A traveller who has a user account. 2. A travel agency who uses their user account to search flights.

Attributes:

1. **email**: Text, 320 chars The user's email address. This email address is used for communication with the user outside the product.

2. **name**: Text, 50 chars The name of the traveller or travel agency using the account.
3. **newsletter**: Boolean Whether the user wants mass email's from the product or not.
4. **city**: Text, 35 chars The city the user's default origin is set to. This is the standard origin used when performing searches for the user.

AccountData - Virtual window: Product-level requirement described using a data dictionary:
Class:

Field	Value
Email	exampleJohn@gmail.com
Name	John Doe
Utskick	True False
City	New York

EmailConfirmation: The ownership of travellers' email addresses should be confirmed. Domain requirement ensuring valid accounts.

ConfirmationCheck: Email verification links should be sent within 5 seconds and expire after 24 hours.

Log email dispatch times in the system and verify expiration behavior after 24 hours.

CreateAccountEmail: Travellers should be able to create accounts with email addresses and passwords. Productlevel authentication requirement.

: The registration process should take no longer than 15 seconds, including email verification.

Log time when server first receives registration form and time of sending verification email.

EmailChange: Travellers should be able to change their email address. Product-level account management feature.

EmailChangeConfirmationPerformance: Changes to email addresses should be confirmed within 10 seconds via a verification email.

Log time when server first receives email change request and time of sending verification email.

ChangePassword:Travellers should be able to change their passwords. Product-level security feature.

CheckChangePassword: Password changes should be effective within 5 seconds of the server receiving the password change request and require strong passwords (at least 8 characters, uppercase, lowercase, number, symbol).

Verify enforcement of password complexity rules and Ensure old credentials become invalid 5 seconds after post-update.

PasswordReset: Travellers should be able to reset their password if they have forgotten it. Product-level usability requirement.

FastPasswordReset: Reset emails should be sent within 10 seconds of server receiving reset request and expire after 30 minutes.

Log request times and email dispatch times. Attempt password resets after 30 minutes to confirm expiration.

MarketingConsent: Travellers should be asked about receiving marketing emails during account creation. This consent should be editable in real-time in the Account management section. The system should log consent preferences for auditing purposes. Domain-level requirement ensuring compliance with marketing preferences.

> **EmailPreferenceCheck**: Marketing email preferences should be configurable and updated in real-time.
>
> Toggle marketing settings in the Account management section and verify backend logs reflect changes within 5 seconds.

ITMaintainance: IT Maintainance shall be able to handle requests from users with issues relating to their user accounts. **ProductMaintainance**: IT Maintainance shall maintain the system over time by keeping track of faults in the user system.

3.6 System Performance

StatisticsReport: Product manager and IT maintenance can request a statistics report containing number of searches and response times. Goal-level requirement for system analytics.

> **FastStatisticsReport**: Reports should be generated in under 10 seconds for a dataset covering the last 6 months.
>
> Execute test report requests and measure query execution and rendering time.

AnonymizedData: User data should be anonymized for analytics.

> **AnonymizedCheck**: Ensure that all personally identifiable information (PII) is removed or obfuscated before generating reports. Verify through data audits and compliance checks.

DynamicResourceAllocation: The system should allocate computing resources (for example memory and CPU), dynamically based on load to ensure scalability. Goal-level requirement for system scalability.

> **ResourceAllocationCheck**: The system should scale up to handle _ concurrent users without slowdown.
>
> Simulate _ concurrent users with load-testing tools and measure system response time. For example AWS.

UptimeGuarantee: The system should have an uptime of at least 99.9%. Goal-level requirement for system reliability.

> **UptimeMonitoring**: Monitor system uptime and ensure it meets the 99.9% requirement.
>
> Use uptime monitoring tools to track system availability over time.

AutoFailoverMechanism: The system should have an automatic failover mechanism. The system should automatically switch to a backup server in case of a failure. Goal-level requirement for system reliability.

> **FailoverTesting**: Simulate a system failure and verify that the failover mechanism is triggered within 5 seconds.
>
> Use automated tests to simulate system failures and measure failover time.

APIIntegrationSupport: The system should support integration with external APIs. Goal-level requirement for system extensibility.

APIIntegrationCheck: Ensure that the system can integrate with at least the necessary external APIs without performance degradation.

Test system performance with and without API integrations to measure impact.

3.6.1 Data Model

DataModel: The product level data requirements are represented through a E/R diagram in Figure 2, it shows the data used

for the flight searches and the user accounts in the system.

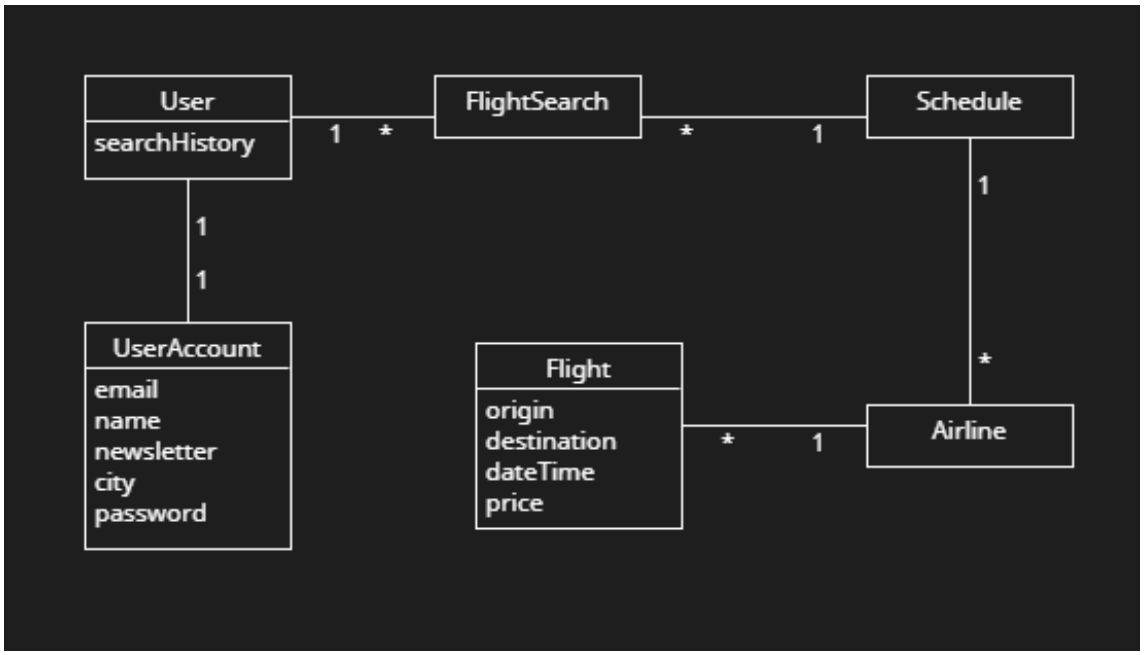

Figure 2: Entity/Relationship data model

3.7 Others

MultipleTicketPrices: Travellers and Travel Agencies can see prices for multiple tickets. Product-level requirement enabling comparison of ticket prices.

FastMultiTicketPrices: The system should allow searching for up to 10 passengers simultaneously without affecting response time. Test by searching for 10 tickets and verifying that all prices are displayed without errors.

Compare response times of searches with 1 vs. 10 passengers under normal and peak loads.

MultiFlightTrips: Travellers and Travel Agencies can find prices of multi-flight trips. Product-level requirement for multi-leg trip pricing.

PriceRetrieval:The system should retrieve multi-flight prices in less than 5 seconds, considering layovers and airlines.

Run tests for complex multi-leg searches, logging response times for different scenarios.

MarketingChannels: Marketers can send marketing emails to travelers who have explicitly opted in through the account management settings. Product-level marketing function.

CheckMarketing: Marketing emails should be delivered to 95% of recipients within 1 minute of sending.

Track email delivery timestamps for mass campaigns and compute success rate over time.

B2BFindFlights: Travel agencies can find flights on behalf of clients. Goal-level requirement supporting B2B services. The agencies use the same type of account as travellers.

AdProviders: The product shall display ads from Ad Providers once every _ minutes.

The following Quality Grid, Table 3, shows the quality characteristics for the system and how we have prioritized them for our application.

	Critical	**Important**	**As usual**	**Unimportant**	**Ignore**
Operation					
Integrity/security			x		
Correctness			x		
Reliability/availability	1				
Usability	3				
Efficiency		2			
Revision					
Maintainability			x		
Testability			x		
Flexibility			x		
Transition					
Portability				x	
Interoperability		4		4	
Reusability					x
Installability		5			5

Table 3: Software Quality Characteristics

Quality Grid Explanations

1. It is critical that the system has high availability since it can be used by anyone across the world, meaning we have the potential to gain customers at any time of the day. If availability is below 99%, the company loses money, and key stakeholders, such as airlines, are negatively impacted.
2. If the system cannot filter or return proper search results quickly, users may grow impatient and leave for a competitor.
3. The system must be easy to navigate and use since our customer base includes anyone willing to travel. Users will have a wide range of experience using websites. Since our differentiation lies in ease of learning we deem it critical that the usability and ease of learning needs to meet our quality requirements.
4. The system is not required to interact with other systems like file transfers or specific hardware. However, some components need to cooperate with external systems, such as our ad

provider or direct links to airlines for flight booking.

5. Since the system is web-based on computer workstations, installation is unnecessary. However, for the mobile version, an app must be installed, which should be easy to download from platforms like the App Store or Google Play.

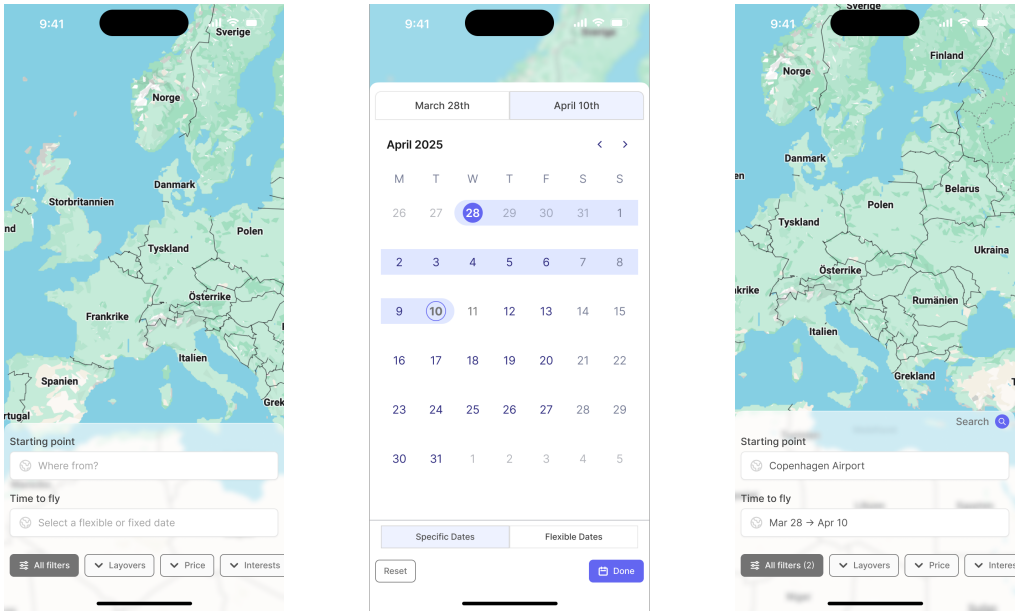

Figure 3: User interface showing the search page, dynamic date picker and filters.

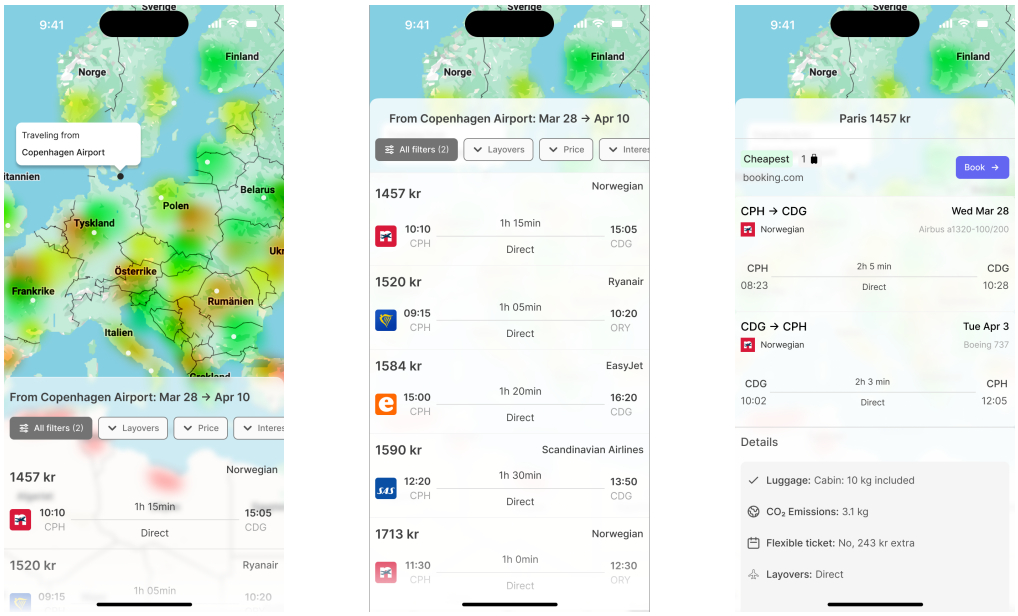

Figure 4: User interface showing the search results page with a list of flights and a heatmap.

4 Release Plan & Prioritization

The release plan is created using reqT, an open source requirements engineering tool that provides a structured approach to requirements modeling and release planning. The tool uses a flexible requirements modeling language that connects entities (like Features) with relations and attributes, enabling systematic analysis and prioritization. Our prioritization strategy uses multiple methods: Our prioritization strategy combines Benefit-Cost Analysis and the MoSCoW prioritization technique to ensure a balanced approach to feature selection.

Benefit-Cost Analysis: Features are evaluated based on their benefit-cost ratio, where benefits represent the estimated business value and costs represent implementation effort. The benefit values are determined by analyzing the priorities of our key stakeholders:

- Travelers (Primary Users): Emphasis on usability, search capabilities, and booking convenience
- Airlines: Focus on integration capabilities, data security, and business operations
- Travel Agencies: Prioritizing B2B features, API support, and comprehensive flight information

The prioritization of stakeholders was created based on our expectations and insights gathered from stakeholder analysis. In the ReqT tool, we prioritized users first, followed by airlines, and lastly travel agencies, to ensure that the most critical needs are addressed effectively.

Priority Classification: Independent of the benefit-cost analysis, we apply the MoSCoW prioritization technique to classify the features into four categories:

- Must-have: Essential features critical for core functionality
- Important: Features that provide significant value but aren't critical
- Nice-to-have: Features that add value but could be excluded

The release planning is formulated as a constraint problem, considering:

- Dependencies between features
- Required implementation order
- Resource allocation across development and testing
- Stakeholder priorities
- Technical constraints

4.1 Feature Dependencies

The implementation order is heavily influenced by technical and logical dependencies between features. Key dependencies include:

- Core Infrastructure Dependencies:
  - SecureDataEncryption must be implemented before AccountCreation
  - APIIntegrationSupport enables B2BFindFlights
- Search Feature Chain:
  - SearchCheapFlights is required for ShowFlights and ListCheapFlights
  - DateRangeMap must precede DateRangeSearch
  - SearchCheapDestinations precedes ListCheapFlights
- Account Management Flow:
  - AccountCreation is prerequisite for:

        * SaveFavoriteCity
        * EmailChange
        * ChangePassword
        * PasswordReset
    - CreateAccountEmail precedes EmailConfirmation
- Advanced Features:
    - ShowFlights is required for:
        * ListFlightsFromMap
        * MultipleTicketPrices
    - MultipleTicketPrices enables MultiFlightTrips

These dependencies have been carefully considered in the release planning to ensure a logical and technically feasible implementation sequence.

While features with the highest benefit and lowest cost are generally preferred, the main focus is on distributing development and testing resources efficiently to enhance productivity and accelerate time to market. Releases 4, 5, and 6 emphasize structured resource allocation to optimize feature delivery.

4.2 Release R4

Benefit: 1000
Cost: 97

Feature	**Benefit**	**Cost**	**Priority** independent from benefit/cost
SecureDataEncryption	88	15	Must-have
MarketingChannels	86	4	Important
UptimeGuarantee	82	9	Must-have
AirportSelection	80	6	Must-have
AnonymizedData	73	10	Must-have
ListFlightsFromMap	73	5	Important
AutoFailoverMechanism	71	11	Must-have
B2BFindFlights	67	2	Important
ListCheapFlights	61	3	Important
DynamicResourceAllocation	54	8	Important
StatisticsReport	51	8	Important
RegularUserFeedback	46	2	Important
CreateAccountEmail	38	4	Must-have
GeolocationService	34	2	Nice-to-have
MarketingConsent	34	1	Must-have
PasswordReset	32	4	Must-have
ChangePassword	28	3	Must-have

4.3 Release R5

Benefit: 465
Cost: 55

Feature	**Benefit**	**Cost**	**Priority** independent from benefit/cost
LawCompliance	89	13	Must-have
ShowFlights	74	7	Must-have
MultipleTicketPrices	72	7	Important
SearchCheapDestinations	68	9	Important
AccountCreation	47	7	Must-have
EmailConfirmation	43	5	Must-have
SaveFavoriteCity	43	3	Nice-to-have
EmailChange	29	4	Important

4.4 Release R6

Benefit: 482
Cost: 55

Feature	**Benefit**	**Cost**	**Priority** independent from benefit/cost
HeatmapSearch	101	18	Must-have
SearchCheapFlights	84	10	Important
APIIntegrationSupport	80	8	Important
DateRangeMap	79	5	Nice-to-have
MultiFlightTrips	78	6	Important
DateRangeSearch	60	8	Important

The MoSCoW prioritization decisions were driven by several key factors. Must-have features primarily focus on essential security (SecureDataEncryption), compliance (LawCompliance), and core user functionality (ShowFlights, AccountCreation) that form the foundation of the flight booking system. Important features were classified based on their significant contribution to user experience and business value, such as search capabilities (SearchCheapFlights) and pricing features (MultipleTicketPrices), which enhance the platform's competitiveness but aren't critical for basic operation. Nice-to-have features like GeolocationService and SaveFavoriteCity represent convenience enhancements that improve user experience but can be implemented later without compromising core functionality.

4.5 Release R7 - Quality Assurance

After release R6, a minimal viable product will be available with all core functionalities implemented. Release R7 focuses on comprehensive quality assurance, including:

- Systematic verification of all quality requirements
- Usability testing and optimization
- Accessibility compliance verification
- Performance testing and optimization
- Security auditing and hardening
- Integration testing across all features
- User acceptance testing
- Documentation completion and validation

Also, ads will be implemented in this release. This dedicated quality assurance phase ensures the product meets all specified standards and provides a robust foundation for future enhancements. During this release, the team will actively work on implementing and refining quality-focused features while conducting the comprehensive testing and validation processes. This parallel approach

ensures that quality is built into the product rather than being treated as just a final verification step.