0. Brief Outline of the Project

The aim of our innovative project is to act as a one stop shop for everything from Preston to Lancaster, boasting dynamic journey routes, and expansive tourist attraction options made easy for the customer. Our app will not only act as a dictionary for users to find attractions, but will actively invite them to popular attractions, making the search process for fun activities as quick as possible. At its core, the app is a well-connected journey planner and directions tool styling powerful navigation technologies to aid users in their everyday activities.

The app will provide alternative routes in case of unexpected disruptions. The system will provide high user security and a wide range of transport options available for the user to choose.
Existing commercial applications may exclude small providers, involve advertising or fees, or prioritise poorly connected interchanges which may lead to a frustrating experience.
Ultimately the project seeks to increase public transport usage by improving reliability, safety and user experience.

1. Feature Comparison

Features	Google Maps (PCMag UK, 2024)	Apple Maps (PCMag UK, 2024)	Waze (PCMag UK, 2026)	TomTom AmiGo (PCMag UK, 2024)	Stagecoach (StageCoach, 2019)
Driving Navigation	✓ Yes – Driving navigation, live traffic, turn-by-turn guidance, lane guidance, speed alerts, route recalculation.	✓ Yes – Driving navigation, CarPlay, live traffic, lane guidance, offline maps.	✓ Yes – Driving navigation, live traffic, hazard alerts, speed alerts, route alternatives.	✓ Yes – Driving navigation, live traffic, lane guidance, speed alerts, offline maps.	✗ No – Bus journey planning only; no driving navigation.
Cycling	✓ Yes – Cycling routes, elevation, bike lanes, turn-by-turn guidance.	✓ Yes – Cycling directions, lane preferences, elevation, traffic-aware routing.	✗ No – Car navigation only.	✗ No – Driving only; no cycling navigation.	✗ No – Bus journey planning only; no cycling navigation.
Street Panoramas	✓ Yes – Street View 360° imagery.	✓ Partial – Look Around panoramas in select cities.	✗ No – No street-level imagery.	✗ No – No street-level imagery.	✗ No – Bus map only; no street-level panoramas.
3D Imagery	✓ Yes – 3D buildings and terrain.	✓ Yes – 3D city models and landmarks.	✗ No – 2D map interface only.	✗ Limited – Basic 2D map, no 3D models.	✗ No – Flat bus route map only.
Map Downloads (Offline Use)	✓ Yes – Offline map downloads and navigation.	✓ Yes – Offline maps supported.	✗ No – Requires internet connection.	✓ Yes – Offline map downloads supported.	✗ No – No offline maps; requires internet for routes and tracking.

Browser-Based Version	✓ Yes – Full web version with route planning and Street View.	✓ Limited – Web version for basic maps and directions.	✓ Limited – Browser version for planning, not navigation.	✗ No – Mobile app only.	✓ Yes (Limited) – Website for planning and tickets; no live navigation.
Walking	✓ Yes – Walking navigation, turn-by-turn guidance, accessibility info.	✓ Yes – Walking navigation, turn-by-turn, AR directions, accessibility.	✗ No – Driving only; no walking navigation.	✗ No – Driving only; no walking navigation.	✓ Limited – Walking to/from bus stops only; no full walking navigation.
Bus	✓ Yes – Bus navigation, real-time info, transfers, alerts.	✓ Yes – Bus navigation, real-time and scheduled info, transfers, accessibility.	✗ No – No bus or public transport navigation.	✗ No – Driving only; no bus navigation.	✓ Yes – Stagecoach bus navigation, live tracking, tickets, service updates.
Train	✓ Yes – Train navigation, real-time info, transfers, disruptions.	✓ Yes – Train navigation, schedules, transfers, real-time updates.	✗ No – No train navigation.	✗ No – Driving only; no train navigation.	✗ No – Bus services only; no train navigation.

2. Software Requirements

2.1 Functional Requirements

The requirements are defined around clear, single responsibilities, ensuring that each functional requirement maps to a distinct system capability. This supports the Single Responsibility Principle (SRP) by preventing overlapping or ambiguous requirements. Requirements are also written in a way that allows new functionality to be added without altering existing ones, aligning with the Open–Closed Principle (OCP).

Functional requirements: The specific tasks that the component must perform and the expected outcomes.

- **Integrated Ticketing & Booking**: The system shall allow users to book and pay for train, coach, and taxi tickets in one simple payment. (thePlanner, 2026))
- **Journey Planning**: The system shall provide a route planner that defragments travel data to offer efficient or cost-effective routes based on user input. (TFL, 2019)
- **Advanced Navigation**: The system shall provide the ability for users to easily add stops to their journeys and save "Favourite Routes", and customise routes based on route preference. (Maps, 2019)
- **Reporting & Crowd Data**: The system shall allow users to report faults (e.g., traffic, accidents), crowding levels, and station accessibility issues.
- **Journey Sharing**: The system shall enable users to share their real-time journey progress with friends and family.
- **Past Journey Retrieval**: The system shall allow users to view details of journeys that have already commenced.
- **Financial Services**: The system shall apply Rail Card discounts and provide a reimbursement system (Delay Repay) for significant delays.

- **Notifications**: The system shall send SMS notifications for route alterations or delays.
- **TripAdvisor**: The system shall integrate with TripAdvisor to provide lodging or tourist attractions
- **Security & Privacy:** The system shall protect user data and transactions through authentication, encryption, and secure access controls.

2.2 Non – Functional Requirements

Non-functional requirements: The performance, reliability, security, and usability requirements of the component.

- **Performance (Speed)**: The system should utilise an advanced caching system and Content Delivery Network (CDN) to provide "second-by-second" updates.
- **Availability (Low Bandwidth)**: The system should ensure travel information remains accessible even in areas with low bandwidth or poor internet connectivity.
- **Security**: The system should undergo a formal security audit (estimated cost £4k-£5k) to ensure user data safety and usability.
- **Maintainability**: The Delay Repayment system should be allocated £20,000 per year for specialized maintenance and upkeep.
- **Infrastructure Reliability**: The system should be hosted on a planned Azure server to ensure stability and support for passenger communications.
- **Usability (Accessibility)**: The app should be designed with "App Accessibility" as a core pillar, ensuring it is usable for all travellers, especially the older demographic.

2.3 Stakeholders

Public Transit Authorities (PTA): Representative who ensure integrated ticketing complies with fare rules and Rail Card policies; support accurate data defragmentation
Safety & Accessibility Advocate (SAA): A representative who ensures accessibility features reflect real-world conditions disabilities; their role is to ensure the "Accessibility Reporting" and "App Accessibility"
System Developer (SD): A lead engineer with expertise in implementing advanced caching and API integration on Azure; manages development.
Security Auditor (SA): A third-party specialist who verifies encrypted customer data storage and assesses web app safety and usability.
Cloud Infrastructure Engineer (CIE): A specialist responsible for the **Azure server** environments and **Content Delivery Network (CDN):** Their role is to ensure the backend scales to handle "second-by-second" updates and remains cost-effective within the £17k annual budget.
Database Architect (DA): A stakeholder focused on the "Defragmentation Engine"; their role is to unify transport data into a single, searchable "Route Object."
API Integration Partner (AIP): External providers like **TripAdvisor** or **Transit Data Aggregators**; they are stakeholders whose system's uptime provide lodging and real-time route alerts.
DevOps/SRE Manager (Site Reliability): Maintains system reliability and automated **delay repayment maintenance** within the £20k **Delay Repayment maintenance budget.**
Compliance & Privacy Officer (CPO): Ensures GDPR and financial regulation compliance for encrypted customer and Rail Card data.

3. Risk Table

Development Risks				
Risk	**Likelihood**	**Effect**	**Strategy**	**Sources**
Loss of data due to unforeseen circumstances or issues	Moderate (Around 25 %)	Catastrophic	Avoidance / Contingency	*(Software Projects Don't Have to Be Late, Costly, and Irrelevant. (2024, April 25). BCG Global.)*

Description: Projects may experience hurdles during development resulting in faulty code or loss of data due to several factors, such as team inexperience. As such it is important to mitigate this by creating constant backups and ensuring all code is understandable.				
Improper implementation of API's	High (Around 50 %)	Serious	Avoidance	*(7 Reasons Why Businesses Are Failing to Secure Their APIs - Cybersecurity Magazine,* 2022*)*
Description: Many projects fail to properly implement and manage any API use, leading to more errors with API calls and more work for developers in charge with maintaining systems. To avoid this, it is important that the API implementation is made to specification and is well documented.				
Security issues found in early penetration testing	Extremely High (Around 80 %)	Catastrophic	Minimisation / Contingency	*(30 SSCS statistics that matter for software security teams,* 2024*)*
Description: Most projects will experience security problems in development that can be catastrophic to the final design. As such is it critical that these issues be researched into and solved, or for more experienced developers to be brought in using overhead.				
Post Development Risks				
Risk	**Likelihood**	**Effect**	**Strategy**	**Sources**
Inaccurate or outdated timetable data from operators	Moderate (Around 20 %)	Serious	Minimisation / Contingency	(*Using Third-Party Applications on Devices*, n.d.)
Description: Leads to frustration or missed connections to users. Automate data updates (via crons) and notify users of possible alternative routes				
User privacy and data security issues	Low (Around 10 %)	Catastrophic	Minimisation / Contingency	(*Secure Development and Deployment Guidance*, 2018)
Description: The app has access to the users' location to provide them with personalised information, as well as personal data such as phone numbers. A security leak may happen. To solve this, there must be contingency plans and advanced security				
Poor user adoption / user experience problems	Moderate (Around 25 %)	Serious	Minimisation / Contingency	(*Rail Passenger Numbers and Crowding on Weekdays (RAI02)*, n.d.)
Description: Poor user adoption leads to misrepresentative analytics for features such as crowding. To solve this, we will scale data based on local market share				
Unexpected transport delays and/or cancellations	Very High (Around 75 %)	Serious	Contingency	(*Home Page | Office of Rail and Road*, n.d.)
Description: Delays and cancellations. According to the Office of Rail and Road (ORR)In 2024/2025 around the 15% of the trains in the UK arrived late. Real-Time information is important, alternative routes will be shown to the user.				
Legal or contractual issues with operators	Low (Around 10 %)	Serious	Avoidance / Contingency	(*Using Third-Party Applications on Devices*, n.d.)
The applications relays on data provided by big / small transport operators. Each of them has their own legal frameworks. We will make sure we abide by TOS				

Accessibility issues for vulnerable users	Moderate (Around 25 %)	Serious	Minimisation / Contingency	World Wide Web Consortium. (n.d.).
As the app will be used by many diverse profiles, either with visibility, mobility and / or technology understanding issues, we need to ensure that the app accomplishes the WCAG compliance and the W3C frontend basics. The app should also provide the users with these disabilities with a solution in accordance with their needs (e.g. A station without elevator)				

4. User Stories

Journey Planning & Route Management

As a passenger…

- I want to plan journeys across multiple transport operators so that I can travel without switching between apps.
- I want transport information from all operators in one place so that journey planning is simple.
- I want bus and rail data combined so that journeys across transport modes are seamless.
- I want multi-leg journey planning so that I can reach destinations requiring several connections.
- I want routes to favour well-served hubs so that I reduce the risk of getting stranded.
- I want routes to avoid unreliable transfers so that journeys are safer.
- I want journey recommendations so that I can choose the fastest or most reliable option.
- I want suggestions for better hubs or routes so that I avoid remote stops.
- I want to be able to change my route once I have started it, including adding additional stops.
- I want stops easily searchable or selectable, so planning is fast.
- I want to see my travel history so I can repeat journeys easily.

Live Travel Information & Smart Predictions

As a passenger…

- I want to see live bus locations so that I know when my bus will arrive.
- I want live train departure information so that I can adjust my travel plans.
- I want delay updates so that I am not surprised by disruptions.
- I want disruption alerts so I can change plans early.
- I want reminders before departure, so I do not miss transport.
- I want predictions based on historical delays so that arrival times are more accurate.
- I want predicted arrival times, so delays are anticipated.
- I want occupancy or crowding predictions so I can choose less busy services.

Stops, Stations & Navigation

As a passenger…

- I want to search for nearby stops and stations so that I can quickly find transport options.
- I want platform and facility information so that I know where to go when I arrive.
- I want to view transport routes on a map so that I understand my journey visually.
- I want live vehicle positions on the map so that I can time my arrival at stops.

Reliability & Safety

As a passenger…

- I want the system to avoid isolated connections so that I do not wait in unsafe locations.
- I want alternative routes suggested when delays occur so that I still reach my destination.
- I want road traffic information so that I know if buses may be delayed.
- I want weather information so that I can plan travel safely.

Ticketing & Payments

As a passenger…

- I want to buy tickets in the app, so I don't need other services.
- I want railcard discounts applied automatically so I pay the correct price.
- I want automatic delay compensation claims, so refunds are easy.
- I want to share journey details or tickets with others so we can travel together.

Accessibility & Inclusivity

As a passenger…

- with mobility needs, I want to see step-free stations so that I can travel independently.
- I want clear connection times so that I can make transfers comfortably.
- I want to report accessibility problems so operators can fix them.
- with disabilities, I want the app usable with screen readers and high contrast so I can travel independently.

Community & Updates

- As a passenger, I want users to report disruptions so information updates faster.
- As a passenger, I want community-confirmed updates, so reports are reliable.

Journey Convenience

- As a passenger, I want to book taxis from the app so that I can complete journeys when public transport ends.

Low Bandwidth & Performance

- As a passenger, I want the app to work in low signal areas so that I can still access travel info.
- As a passenger, I want lightweight data mode so that maps and updates load quickly.

4.1 Infrastructure & Technical Stories

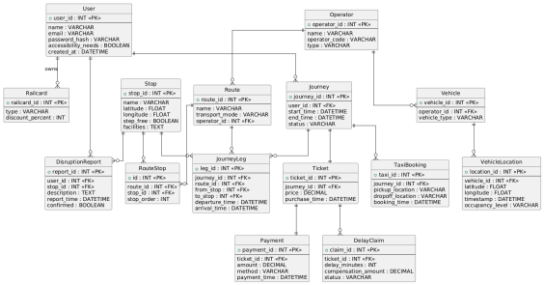
Content Delivery Network (CDN)

- As a user, I want data delivered from nearby servers, so app loading is faster.
- As a system owner, I want scalable delivery so peak usage does not crash the system.

Admin / Operator Stories

- As a system admin, I want to monitor service data feeds so that broken feeds are detected quickly.
- As an admin, I want system analytics so usage patterns can improve planning.

5. Database Design



The image located above can be found in greater quality in the GitHub repository linked in the appendix.

The database is designed to support a transport system which is a collective of buses, trains, and taxis even. The system enables users to plan journeys, purchase ticket, track vehicles, and handle disruptions and delays with ease, even being forwarded the opportunity to make delay repay claims directly in the app.

A relational database model was chosen because the system requires strong data consistency for clear relationships between entities (users, journeys, routes), and transactional integrity for payments and claims.

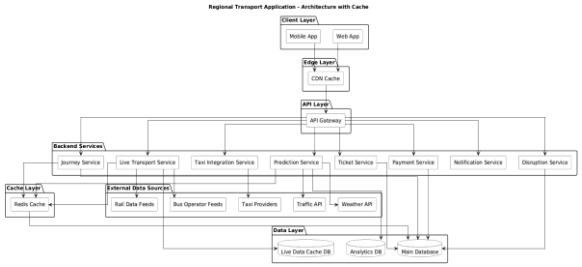
The core User and account entities store registered users of the system. The key attributes are as follows: user_id (PK); email; password_hash which stores hashed passwords for security, each password is also hashed with a salt as to make them less suspetible from rainbow table; accessibility_needs which indicates whether accessibility features are required; and a created_at timestamp which stamps when the account was created.

The rationality behind the design to separate users from journeys is to allow a single suer to plan and store multiple journeys.

The database schema applies SRP by ensuring each table models one real-world concept (e.g. Journey, Ticket, Payment). Relationships are defined through foreign keys rather than duplicated data, supporting OCP by allowing schema extension without modification. This structure also aligns with DIP, as application logic is decoupled from physical data storage.

Caching responsibilities are isolated from core business logic, adhering to SRP. Backend services interact with cache layers through abstractions rather than direct implementations, supporting DIP. This allows caching strategies (e.g. Redis, CDN) to change without impacting service logic

6. API Design



The image located above can be found in greater quality in the GitHub repository linked in the appendix.

The system uses a layered API architecture to support a regional transport application across both web and mobile clients. The design follows a gateway + service-based approach, where requests are routed through a single API entry point (API Gateway) and then handled by specialised backend services (e.g., journeys, live transport, payments).
The architecture facilitates: scalability, maintainability, performance, and resilience. The Client layer supports both mobile app, and if we choose to implement it, a web app too, of course that is subject to future requirements, however for pragmatic purposes, we like to keep our, and our client's options open. External APIs (rail feeds, taxi providers, weather services) are accessed via dedicated integration services, ensuring SRP. This shields the core system from changes in third-party APIs, aligning with OCP and DIP by depending on internal abstractions rather than external implementations.

7. Interaction Diagram

The interaction diagram can be found in the appendix; due to its length it does not fit here.
Or for quick access, you can find it in the link below
Link to Diagram

Scenario 1: Journey Planning & Route Optimisation

This scenario shows how a user requests journey options via the mobile application. The request is routed through the API gateway to the journey service, which first checks cached route data before querying live transport services in the case of a cache miss. We have a very advanced caching system to minimise bandwidth usage as we know the importance of speed when it comes to traveling applications. Accessibility information data is incorporated to produce optimised routes and to ensure accurate and inclusive journey planning.

Scenario 2: Advanced Journey Customisation

Users can modify journeys by adding and removing stops, and updates are made to the database and cache accordingly.

Scenario 3: Integrated Ticketing & Payment

Once a route is confirmed, the ticket service creates a ticket record and prepares it for payment. The payment is then processed, discounts applied where appropriate, and a notification is triggered. All of these stages are separated, ensuring single responsibility principles, which ensures security of financial data.

Scenario 4: Real-Time Journey Tracking & Notifications

Throughout the journey, the system continuously retrieves live vehicle location and occupancy data. Updates are streamed through the live transport service and is stored in the live data cache, enabling real-time map updates and ETA calculations. To save on bandwidth, updates are only sent when a change is detected. In the event of a very important trip and time update, SMS messages are sent to the user as they are quicker and less signal demanding.

Scenario 5: Journey Sharing

This scenario allows users to share their journey progress with others. A secure share token is generated and stored, enabling recipients to view live journey updates without accessing the user's account. This interaction demonstrates controlled data sharing and minimal exposure of sensitive data.
The layered architecture enforces SRP by separating concerns between client, API, services, and data layers. High-level modules (clients and gateway) do not depend on low-level details (databases or external feeds), which directly applies Dependency Inversion. This results in a flexible, testable, and scalable system.

Scenario 6: Disruption Reporting & Crowd Data

Users can report disruptions such as station issues, overcrowding is calculated as follows: number of users we have on a given train or route, divided by our percent market share, times by 100 which will give us an estimate of how crowded a train is. This is of course an estimate and assumes that our market share is reflected by passengers on the train. This data is stored in the main database and used to update disruption caches. The notification service then informs affected user.

Scenario 7: Delay Compensation

When delays exceed a predefined threshold, users are eligible to claim compensation if the so choose. Data pertaining to the trains are stored to check for eligibility of delay repays. This data is stored for as long as it is needed until the period where users can make a claim expires.

Scenario 8: TripAdvisor Integration

This scenario integrates third-party tourist information. When users request nearby attractions, the system queries the TripAdvisor API and stores the results. The integration will be designed and formatted to minimise work for the users. The goal is to make it feel like users are being invited to attend attractions instead of having to seek them out. We want to minimise the work the user has to do, as every hoop and hurdle is something else stopping our users from enjoying their trip.

Scenario 9: Multi-Modal Journey (Train + Taxi)

The journey service system searches across multiple operators and transport modes retrieving the most optimal modes of transport which will be based on users' preferences, such as price, least changes, quickest, etc. This scenario is the bread and butter of the program; it is what gets users from A to B in the most appropriate way.

8. Data Ownership & Governance Matrix

The following table provides a domain-oriented overview of the system's core data entities, showing how responsibilities are distributed across services, storage technologies, and event flows. Each domain is owned by a single service, reinforcing clear ownership and alignment with the system's service-based architecture. The data aims to demonstrate how all entities and components are cleanly separated while remaining coordinated through well-defined events and data boundaries, fully adhering to single responsibility principles.

Domain	Data	Owner	Storage	Events	Notes
Identity	Users	User Svc	RDBMS	UserCreated, UserUpdated	Source of truth
	Railcards	User Svc	RDBMS	RailcardAdded, RailcardExpired	Read by Ticket Svc
Journey	Routes	Journey Svc	RDBMS	RouteUpdated	Static reference
	Stops	Journey Svc	RDBMS	StopUpdated	Shared read-only
	Schedules	Journey Svc	RDBMS	ScheduleUpdated	Timetable SoT
	Journeys	Journey Svc	RDBMS	JourneyCreated, JourneyCancelled	Ticket validation
	Journey Legs	Journey Svc	RDBMS	JourneyLegUpdated	Internal structure
Live Ops	Vehicles	Live Transport Svc	RDBMS	VehicleRegistered	Static ops data
	Operators	Live Transport Svc	RDBMS	OperatorUpdated	Ownership data
	Vehicle Locations	Live Transport Svc	Redis	VehicleLocationUpdated	Ephemeral
Ticketing	Tickets	Ticket Svc	RDBMS	TicketPurchased, TicketCancelled	Immutable
	Delay Claims	Ticket Svc	RDBMS	DelayClaim*	Depends on disruption
Payments	Payments	Payment Svc	Secure DB	PaymentCompleted, PaymentFailed	PCI-DSS
Disruption	Disruption Reports	Disruption Svc	RDBMS	Disruption*	Impacts journeys
Integrations	Taxi Bookings	Taxi Svc	RDBMS	TaxiBooked, TaxiCancelled	External boundary
	TripAdvisor Data	TripAdvisor Svc	RDBMS / Redis	Trip*	External boundary
Analytics	Analytics Events	Prediction Svc	Analytics DB	N/A	Append-only
	Predictions	Prediction Svc	Analytics DB	PredictionUpdated	Derived only

9. Acceptance Tests

ID	Test Description	Test Steps	Expected Results	Pass/Fail
1	Verify a user can plan a complete journey in one place	User opens the app. User enters start and destination.	App loads successfully to the home screen. Route options are displayed.	

		User views tickets, delays, and accessibility info.	All information is shown in one place.	
2	Verify integrated ticket purchasing works	User plans a journey. User buys a ticket in the app. User views ticket.	Journey details are shown. Ticket purchase completes successfully. Ticket is stored and available offline.	
3	Verify users receive disruption notifications via SMS	User enables notifications. A delay occurs User opens notification	Settings are saved. Notification is received via SMS. Delay information is clear.	
4	Verify accessibility information and reporting	User searches for a station/bus stop. User views accessibility info. User submits an accessibility report.	Station/bus stop page loads. Step-free access and issues are visible. Report is successfully submitted.	
5	Verify app usability in low bandwidth areas	User opens app on slow connection. User opens saved journey. User opens saved ticket.	Essential content loads. Journey details load. Ticket is accessible offline and online.	
6	Verify users can personalise journeys	User plans a route. Users filter their travel options such as budget range, transport options and time. User saves route as favourite.	Route options appear. Options appear that tailor to their needs. Route is saved.	
7	Verify community travel updates are useful	User opens community updates. User reads an update.	Updates are displayed. Information is clear and helpful.	
8	Verify Delay Repay feature	User selects a delayed journey. User submits Delay Repay claim. User receives confirmation.	Delay details are shown. Claim is submitted successfully. Confirmation message is shown.	
9	Verify railcard support	User adds a railcard. User plans a journey. User edits or removes railcard.	Railcard is saved. Discounted price is applied. Changes are saved.	
10	Verify viewing past journeys	User opens journey history. User selects a past journey. User reuses a past journey.	Past journeys are listed. Journey details are displayed. Journey can be replanned.	

11	Verify crowding prediction data	User plans a journey. User views crowding information. User changes travel time.	Route options are shown. Crowding levels are displayed. Crowding prediction updates.	
12	Verify trip adviser suggestions	User plans a journey. App suggests alternative routes. User selects suggested route.	Journey options appear. Suggestions are shown. Journey updates successfully.	
13	Verify sharing journeys with others	User shares journey. Another user opens shared journey. User views shared ticket info.	Share option works. Journey details are visible. Information is clear despite provider limits.	
14	Verify taxi booking integration	User reaches end of journey plan. User selects taxi booking. User confirms booking.	Taxi option is shown. Pick-up and drop-off are auto-filled. Taxi booking is successful.	
15	Verify app accessibility features	User enables screen reader. User increases text size. User navigates key features.	App is navigable. Layout remains usable. No accessibility barriers present.	

9.1 Business Acceptance Test (BAT)

ID	Business Goal	Acceptance Criteria	Test Scenario	Expected Result	Pass/Fail
1	Multi-operator coverage	Allows user to make journeys from Preston and Lancaster including Blackpool, the Fylde and the Wyre coast.	Plan journey from a rural village to Lancaster	Journey shows all viable connections, including small operators	
2	Safety & Practical Journeys	Avoids isolated stops or errors/features/information that may confuse the user	Have a list of users from ranging demographics test the app	Feedback states that they were happy with the app and that there were no errors	
3	Cost & ads	Free, non-intrusive	User tests full journey planning	No fees, no disruptive ads	
4	Timely info	Reflects schedule updates	Change a timetable entry	App updates suggested routes accordingly	
5	Promotion of public transport	Encourages usage	Compare statistics of then vs now	Increase in public transport usage	
6	Reduce fragmentation of information	Users can plan multi-travel journeys without the user being confused	Test with users	Users are able to experience journeys with zero confusion	

10. UI

Image 1

Image 2
Image 3
Image 1 depicts the interface of a journey with multiple stops. The first of which shows the user being taken on a train, which shows the user what stops they must be on, and what platform the train is on; accessibility information is also shown. The same image also shows two more stages after the train: that being, bus and walking stage. Those icons open in a similar way to the train section.
Image 2 shows how the bus system works, as well as the map, the user is given two options: directions, which takes the user to the stop or location that they have selected; and departures, which shows what buses depart from a given stop.
Image 3 depicts a planned interface for viewing bus timetables within a period at a particular station, as well as depicting the states of the more immediate busses, as to whether they are crowded or available. It also provides a typical option to view later departures as well as being able to favourite the stop for easier viewing later.
More tested and considered images are in the GitHub, found in the appendix.

11. Plan for implementation, milestones and deliverables

Overview

We are going to distribute the milestones in layers

Layer	Predicted impact
Client and Edge	Never before the backend core
API Gateway	Early milestone
Backend Services	Main phase
Cache	Late milestone - Optimization
Data	Early milestone (Optimize in the end)

Plan and implementation

Objectives:

First, the system's architecture is fully designed towards the development of a useful transport application, capable of covering the Lancashire area (including the coastal region). Additionally, some deeper objectives are going to be journey planning, ticketing, real-time updates and accessibility-aware travelling.
Focus will be on delivering a scalable backend and resilient data supported with caching and CDN technologies.

Implementation Strategy:

Core functionality, independent functionality, and functionality which is depended upon by other features will be implemented first to create a secure program infrastructure which can be built upon. All user related functionality will be implemented before analytic functionality.

Milestones and Deliverables

Core Platform and Data Foundations

The term limit that the group has agreed to achieve this is during **weeks 15 - 16.**
This milestone establishes the basics of the project, which will enable all subsequent services to properly work and rely on this core and data.

Implemented components:

API Gateway and routing **-** Journey service and routes logic **-** Integration with rail and bus operators' feeds **-** UI implementation and development

Deliverables:

A backend capable of generating journey plans - Verified data integrity and implementation from transport operators - Internal API documentation

Ticketing, Payments and other features

The term limit that the group has agreed to achieve this is during **weeks 17 - 18.**
20To be aligned with the stakeholders, part of the goal regarding this milestone is to do a fare compliance. In addition, the handling of payment services so that security auditors can be passed, is also considered.

Implemented components:

Ticket and payment service - Railcard discount and fare calculations - Live delay and disruptions services - Notification services, that is, SMS's - Analytics DB for journey and delay tracking

Deliverables:

End-to-end ticket purchase and payment flow - Possible delay repays calculations - Initial demo ready to be released

Enhancements and external integrations

The term limit that the group has agreed to achieve this is during **weeks 19 - 20.**
The purpose of this milestone is to enhance User Experience (UX) by adding services to the application. Moreover, refinement and user's freedom will be essential to achieve and yet, aligned with functional requirements described in the project.

Implemented components:

Taxi Integration Service - Past journey retrieval - Accessibility reporting - UI improvement for Low-bandwidth scenarios

Deliverables:

Application with all the features and requirements - Accessibility validation review

Security and Final Delivery

The term limit that the group has agreed to achieve this is during **week 21.**
To test how secure the application is, we need to have data and backend services first. Then, this milestone objective is to enhance its security and identify possible data disclosures.
Cache and CDN optimization is also implemented in this phase.
Finally, the application is ready to efficiently and effectively work.

Implemented components:

CDN and cache optimisation / tuning - Encryption methods applied to data - Possible bug fixing - Final deployment

Deliverables:

Final codebase and documentation - Deployment and maintenance notes

Progress monitoring

Progress will be monitored by the milestones specified previously. Moreover, communication between the group members will be continuous, with weekly meetings and lab sessions.
Deliverables are validated against both, functional and non-functional requirements, so that system responsiveness is always a main priority.

12. Costs analysis

Documentation on contingency costs (overhead), operational cost (OpEx) and upfront cost (CapEx) for a year of operation. This covers cost of labour, services, hosting and contingency costs. Totalling up to a final figure of

Title	Figure	Cost type	Description

Labour	~£105,000	OpEX	The payment due for staff members working on the project
Service/API	~£50,000-£75,000	OpEX, CapEX	Cost relating to information delivery, SMS, integrations of outside sources
Server/Hosting	~£17,000-£20,000	CapEX	The deficit incurred by hardware/software resources needed for processing.
Risk Associated Costs	~£62,500 (~25% of budget)	Overhead (Links to other associated costs; worst case scenarios)	All potential cost incurred if worse case scenarios occur.
Final Cost	~£250,000		

12.1 Labour Costs

Job Title	Hours	Time Span	Hourly Rate	Total	Number of Employees
Senior Software Engineer	*~150*	Jan26-Apr26	**£50**	£7,500	1
Project Manager	~150	Jan26-Apr26	**£50**	£7,500	1
Data Engineer	~150	Jan26-Apr26	**£40**	£6,000	1
Operation Security/Maintenance Officer	Salary	Jan26-Jan27	**TBD**	~£65,000	1
UI Developer	~150	Jan26-Apr26	**£25**	£3,750	1
Junior Software Engineer	~150	Jan26-Apr26	**£25**	£7,500	2

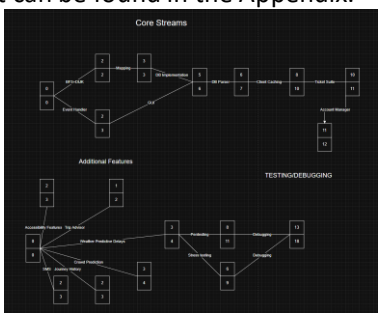
12.2 Service/API Costs

There are numerous major considerations that need to be made when estimating the cost incurred from the services and API's the app will interface with; for fast communications in events of delay or cancellation that effect the end user we will be utilising Azures SMS services as each message sent costs £0.03 we estimate total cost per annum to be ~£9000 (modelled with ~300,000 messages). Trip Advisor allows for integration of their software for free with rate limits per day besides this another limitation is the documentation needed to be approved for the API suite. Rate limits are 10,000 calls per day with a max of 50 per second. For ticket issuing there are two critical API layers involved in the process. Fare retrieval which its associated costs total £5,000 per year and a retail engine which provides the means to sell tickets costing roughly £5,000 upfront with an additional £4,000 in commission. Licensing is also a need and comes at the fee of ~£10,000 per year. Using Microsoft's Copilot will mean we incur a cost of £13.80 per user every month totalling ~£1200 per annum.

13. Gantt and Activity Network

Task	WEEKS						
TIMESPAN	Week 15	Week 16	Week 17	Week 18	Week 19	Week 20	Week 21
CORE FUNCTIONALITY							
Graphical User Interface Components							
Database Components							
Additonal Functionality							
Accessibility Features and SMS							
Predictive Features and Reviews							
Testing/Debugging							

The figure above illustrates how our project has 3 key stages. The construction of the fundamentals of the app, the value added through additional features and the extensive testing of these systems for the launch date. The link to the full chart can be found in the Appendix.



The image located above can be found in greater quality in the GitHub repository linked in the appendix. The figure above Illustrates which tasks are dependent on each other or previous tasks and the worst case and best-case times to complete of the associated stages of development.

Appendix

Contents

GitHub Page:

https://github.com/saamaaraiii/SCC200-Group6-5-

1. Architecture Diagram PNG

https://github.com/saamaaraiii/SCC200-Group6-5-/blob/main/System%20Architecture%20Diagram%20(Chris).png

2. Database Diagram PNG

https://github.com/saamaaraiii/SCC200-Group6-5-/blob/main/Database%20Diagram.png

3. API Design PNG

https://github.com/saamaaraiii/SCC200-Group6-5-/blob/main/API%20Design.png

4. Gantt Chart

https://github.com/saamaaraiii/SCC200-Group6-5-/blob/main/project%20gantt%20Excel.xlsx

5. Activity Network Diagram

Activity Network (it might make you download the image to see it)

2. AI usage:

AI was used in assistance throughout the whole of the work:

The following table details what AI was used for and is not suggestive of the work done by humans.

Section	Tool used	How did you use it	How was the output verified
Database Design	Co-pilot	Was used to generate a plant UML script based on a given input which supplied the AI with context of our project requirements, and features.	Checked over, changes made where necessary
API Design	Co-pilot	Was used to generate a plant UML script based on a given input which supplied the AI with context of our project requirements, and features.	Checked over, changes made where necessary Checked over, changes made where necessary
User Stories	Co-pilot	Was used to research industry standard data ownership models and tables, and was then used to	Checked over, changes made where necessary

		generate a table based on the work another member of this group.	
Software Requirements	Co-pilot	Was used to generate scenarios where differing users might need specific requirements, user stories where then generated based on those scenarios.	Checked over, changes made where necessary
Project Plan for Implementation including Milestones and deliverables	Co-pilot	Was used to give the group an overview about what milestones were more important to achieve. Additionally, it was used to check the complexity of some of the tasks and, based on that, re-structure	Checked over, changes made where necessary
Cost Analysis	Co-Pilot	Copilot was used to gain information pertaining to fare retrieval and ticket issuing as this is not publicly available information. Copilot was also used for estimates of wages.	Compared information to verified sources where applicable

We declare that we have used Artificial Intelligence tools only as described above. All submitted work reflects my own understanding, and I take full responsibility for its accuracy and integrity.

Bibliography

PCMag UK. (2024). *Apple Maps vs. Google Maps vs. Waze: The Best Navigation Apps for 2024*. [online] Available at: https://uk.pcmag.com/gps-navigation/151741/apple-maps-vs-google-maps-vs-waze-the-best-navigation-apps-for-2024.

PCMag UK. (2026). *Waze Finally Delivers 5 Features It First Promised Almost 2 Years Ago*. [online] Available at: https://uk.pcmag.com/gps-navigation/162736/waze-finally-delivers-5-features-it-first-promised-almost-2-years-ago [Accessed 9 Feb. 2026].

www.stagecoachbus.com. (n.d.). *New and Updated Bus Travel App | Stagecoach Bus App*. [online] Available at: https://www.stagecoachbus.com/promos-and-offers/national/stagecoachbusapp.

Home page | Office of Rail and Road. (n.d.). Www.orr.gov.uk. https://www.orr.gov.uk/

Initiative (WAI), W. W. A. (n.d.). *W3C Accessibility Standards Overview*. Web Accessibility Initiative (WAI). https://www.w3.org/WAI/standards-guidelines/#wcag2

Rail passenger numbers and crowding on weekdays (RAI02). (n.d.). GOV.UK. https://www.gov.uk/government/statistical-data-sets/rai02-capacity-and-overcrowding

Secure development and deployment guidance. (2018, November 22). Www.ncsc.gov.uk. https://www.ncsc.gov.uk/collection/developers-collection

Using third-party applications on devices. (n.d.). Www.ncsc.gov.uk. https://www.ncsc.gov.uk/collection/device-security-guidance/policies-and-settings/using-third-party-applications-on-devices

Software Projects Don't Have to Be Late, Costly, and Irrelevant. (2024, April 25). BCG Global. https://www.bcg.com/publications/2024/software-projects-dont-have-to-be-late-costly-and-irrelevant

Kent, J., & Kent, J. (2022, July 25). *7 Reasons Why Businesses Are Failing to Secure Their APIs - Cybersecurity Magazine*. Cybersecurity Magazine - Science Meets Practice. https://cybersecurity-magazine.com/7-reasons-why-businesses-are-failing-to-secure-their-apis/

Vijayan, J. (2024). *30 SSCS statistics that matter for software security teams*. [online] ReversingLabs. Available at: https://www.reversinglabs.com/blog/software-supply-chain-security-by-the-numbers-30-key-stats-that-matter.

Reference listMaps, G. (2019). Get directions & show routes in Google Maps - Computer - Google Maps Help. [online] Google.com. Available at: https://support.google.com/maps/answer/144339.saamaaraiii (2025). GitHub - saamaaraiii/SCC200-Group6-5-: Group 6-5 for group project at Lancaster university. [online] GitHub. Available at: https://github.com/saamaaraiii/SCC200-Group6-5- [Accessed 9 Feb. 2026].StageCoach (2019). Welcome to Stagecoach. [online] @stagecoachgroup. Available at: https://www.stagecoachbus.com/.TFL (2019). Plan a journey. [online] Transport for London. Available at: https://tfl.gov.uk/plan-a-journey/.thePlanner (2026). Sitio bloqueado. [online] Theplanner.co.uk. Available at: https://www.theplanner.co.uk/2026/01/22/ [Accessed 11 Feb. 2026].

Copilot Cost - Microsoft.com. (2025). *Microsoft 365 Copilot Plans and Pricing—AI for Business | Microsoft 365*. [online] Available at: https://www.microsoft.com/en-gb/microsoft-365-copilot/pricing. Accessed 27.1.25

SMS - Prakulka (2025). *SMS pricing - An Azure Communication Services concept document*. [online] Microsoft.com. Available at: https://learn.microsoft.com/en-us/azure/communication-services/concepts/sms-pricing?pivots=tollfree [Accessed 12 Feb. 2026].

accessed 27.1.25

Server Cost – Accessed 27.1.25

azure.microsoft.com. (n.d.). *Pricing – Cloud Services | Microsoft Azure*. [online] Available at: https://azure.microsoft.com/en-gb/pricing/details/cloud-services/.

TripAdvisor - Kozinskiy, A. (2020). *Tripadvisor API Access Key: partnership, cost, integration [2024]*. [online] Elfsight. Available at: https://elfsight.com/blog/how-to-get-tripadvisor-api-key/. Accessed 27.1.25