

Technical Report - **Product specification**

# GateMate

Course: IES - Introdução à Engenharia de Software

Date: Aveiro, 24 October 2023

Students: 107418: Bruno Páscoa  
107637: André Almeida Oliveira  
107359: Duarte Carvalho da Cruz  
108796; Sara Figueiredo Almeida

Project abstract: GateMate is centered around the collection, organization, and presentation of flight data obtained through the AvitionStack API. It addresses the high-level business problem of simplifying and enriching the airport experience for travelers.

Table of contents:

[1 Introduction](#)

[2 Product concept](#)

[Vision statement](#)

[Personas](#)

[Main scenarios](#)

[3 Architecture notebook](#)

[Key requirements and constrains](#)

[Architetural view](#)

[Module interactions](#)

[4 Information perspective](#)

[5 References and resources](#)

# 1 Introduction

This project was initiated with the primary objective of developing a fully functional, multi-service application. Beyond the application itself, it serves as a comprehensive learning experience encompassing various aspects of application development, such as user stories, branch management, and agile architecture. This report details our journey in achieving these objectives.

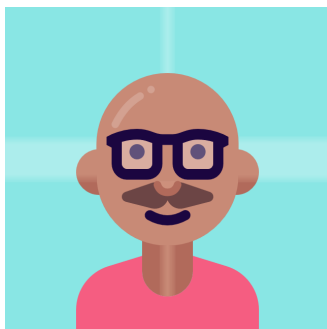
## 2 Product concept

### Vision statement

GateMate is centered around the collection, organization, and presentation of flight data obtained through the AviationStack API. It addresses the high-level business problem of simplifying and enriching the airport experience for travelers. It achieves this by offering a comprehensive solution for gathering, organizing, and presenting real-time flight data. Travelers often face challenges in navigating through airports, keeping track of gate changes, and staying informed about the status of their flights. This application resolves these challenges by enabling users to scan their boarding passes or manually search for their flight, after which they receive up-to-the-minute notifications and directions, ensuring they are always in the right place at the right time.

While several flight-tracking applications exist, our system differentiates itself by focusing on the integration of multiple services, combining AviationStack API data with user-friendly functionalities. Our application is designed to be more than just a flight tracker; it's a travel companion that offers an intelligent, context-aware experience for travelers.

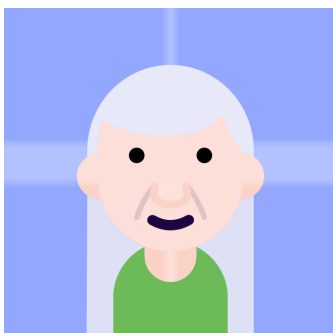
### Personas and Scenarios



**Hank Greens** is a forty-three-year old environmental expert. He loves his job as it allows him to be in contact with nature, which he was passionate about ever since he was a kid.

He has a wife, Daisy and a 5-year-old daughter named Rose.

**Motivation:** as his job requires him to travel by plane often, he wants to use this app to keep track of flights to make sure he doesn't miss his flights.



**Sara Sousa** is a seventy-year-old woman. She used to work as a waitress before she retired. Despite her age, Sara is very active and often goes for jogs in the morning.

Sara's husband died recently, so she travels often to see her son, Matias, who lives abroad.

**Motivation:** Sara has had some trouble tracking flights recently and wants a site that helps her double-check where to go.



**Carlos Andrade** is a 35-year-old marketing manager with a passion for photography and travel. Mark frequently travels for both work and leisure, exploring various destinations with his family.

He values his travel experiences and wants to ensure that every aspect of his journey is as smooth as possible

**Motivation:** Carlos recently suffered a leg injury that limits his mobility. This injury makes it difficult for him to constantly get up to check if his flight's gate has opened, as he used to.

### Scenarios:

**Hank tracks the plane:** Hank is nervous since he doesn't have a lot of time between his arrival and the connection flight. Hank goes to the webpage, goes to "flights", filters by airport and sees that the plane has left way later than he was supposed to. Hank then contacts his flight agency and manages to change his connection flight to a later time.

**Sara subscribes to flight:** Sara is at the airport and doesn't want to miss the flight. So she goes to "flights", searches for the flight and subscribes to it. A few minutes before the flight, Sara gets an email informing her that her plane is about to arrive, so Sara goes to the terminal and gate indicated in the email and catches the flight.

**Mark's Gate Notification Assistance:** Carlos Andrade, who recently suffered a leg injury, is at the airport with his family. He's in a cast and can't easily get up to check the airport information board. Mark uses the application to receive a notification as soon as the gate for his flight opens. This feature allows him to comfortably stay seated and not worry about missing his flight, giving him peace of mind while he recovers from his injury.

## Product requirements (User stories)

- As a User, I want to search and filter for flights so I can get more information about the flight I want.
- As a User, I want to receive a notification when gates open.
- As a User, I want to receive real-time data for a flight so I can know if it's delayed.
- As a User, I want to register and login, so I can subscribe to flights I want to be notified about.
- As a User, I want to be notified when the plane is about to take off, so I don't miss it.
- As a User, I want to be notified when the plane is delayed, so I can reschedule connection flights if needed.



## 3 Architecture notebook

### Key requirements and constraints

<Identify issues that will drive the choices for the architecture such as: Will the system be driven by complex deployment concerns, adapting to legacy systems, or performance issues? Does it need to be robust for long-term maintenance?

Identify critical issues that must be addressed by the architecture, such as: Are there hardware dependencies that should be isolated from the rest of the system? Does the system need to function efficiently under unusual conditions? Are there integrations with external systems? Is the system to be offered in different user-interfacing platforms (web, mobile devices, big screens,...)?

E.g.: (the references cited in [XX ] would be hypothetical links to previous specification documents/deliverables )

There are some key requirements and system constraints that have a significant bearing on the architecture. They are:

- The existing legacy Course Catalog System at Wylie College must be accessed to retrieve all course information for the current semester. The C-Registration System must support the data formats and DBMS of the legacy Course Catalog System [E2].
- The existing legacy Billing System at Wylie College must be interfaced with to support billing of students. This interface is defined in the Course Billing Interface Specification [E1].
- All student, professor, and Registrar functionality must be available from both local campus PCs and remote PCs with internet dial up connections.
- The C-Registration System must ensure complete protection of data from unauthorized access. All remote accesses are subject to user identification and password control.
- The C-Registration System will be implemented as a client-server system. The client portion resides on PCs and the server portion must operate on the Wylie College UNIX Server. [E2]
- All performance and loading requirements, as stipulated in the Vision Document [E2] and the Supplementary Specification [15], must be taken into consideration as the architecture is being developed.>

### Architeturall view

- Discuss architecture planned for the software solution.
- include a diagram

### Module interactions

- explain how the identified modules will interact. Use sequence diagrams to clarify the

interactions along time, when needed

→ discuss more advanced app design issues: integration with Internet-based external services, data synchronization strategy, distributed workflows, push notifications mechanism, distribution of updates to distributed devices, etc.>

## 4 Information perspective

<which concepts will be managed in this domain? How are they related?>

<use a logical model (UML classes) to explain the concepts of the domain and their attributes>

## 5 References and resources

<document the key components (e.g.: libraries, web services) or key references (e.g.: blog post) used that were really helpful and certainly would help other students pursuing a similar work>