MonTraveller
Vision Document
Version <1.0>

TABLE OF CONTENTS

Vision Document
MonTraveller

1. INTRODUCTION

This document describes the vision behind the MonTraveller app; why it exists and what problems it aims to solve. It includes an analysis of both stakeholders needs and the product's market. Finally, it defines high-level product features and requirements of the system to be taken on in the future. The MonTraveller app serves as a way to connect Montrealers to their city and does so by offering concrete solutions to everyday problems. The key areas affected by the app include: transportation, social events, shopping and social media. Having all these areas taken into account by the application makes up for an ecosystem like no other where users can learn, connect and share all in one place.

2. POSITIONING

This section of the Vision Document summarizes the problem being solved with the development of the MonTraveller app and explains the position that this mobile application will fill in the marketplace.

2.1. Problem Statement

The problem of	Not knowing who or what is available around you.
Affects	Citizens, employees in catering, hotels, transportation, entertainment.
The impact of which is	Montrealers do not know what to do in their free time and where to find up-to-date information about events in the city.
A successful solution would be	An active user base that knows Montreal in and out and that can connect with its environment and most importantly others within it.

2.2. Product Position Statement

For	Montrealers and tourists in Montreal
Who	are searching for an activity to take part in.
MonTraveller	is a mobile application.
That	Offers users something to do that they like.
Unlike	● Applications that don't have full integration with transportation and accommodation. ● Event listing apps that only include community-run events. ● Hotel apps that do not include traveling costs.
Our product	Offers an exhaustive assortment of activities to do, including reviews and transportation.

3. STAKEHOLDER DESCRIPTIONS

In this section of the Vision Document, we identify the various stakeholders involved in our software project, the MonTraveller app. Next, we describe each stakeholder and list their key responsibilities with regard to the system being developed. We then conclude the section by detailing the working environment of the target user.

3.1. Stakeholder Summary

Name	Description	Responsibilities
Organization	The MonTraveller Organization is composed of individuals who work for the application. It includes the product owners and company employees.	● Achieve successful development, deployments and maintenance of the MonTraveller app ● Become a profitable and competitive local business ● Ensures market demand for existing and future features
Software Developers	They are technically knowledgeable professionals who will be designing, developing, testing, and maintaining the MonTraveller app.	● Translate user needs into system features ● Develop and deploy the mobile app ● Fix system bugs ● Deliver system updates
Management Team	They are resourceful professionals who take big project decisions and ensure communication between members of the project team.	● Risk Analysis ● Project milestone estimates ● Interdepartmental communication ● Decision making authority ● Monitors application analytics
Marketing Team	They are the company employees that will use media and marketing strategies to introduce the MonTraveller app to the intended end-users.	● Spread awareness of the MonTraveller app ● Maintain high rates of app usage ● Increase number of end-users ● Liaise with app sponsors

End-Users	They are the people who will be using the MonTraveller app and its many functionalities.	● Key influence on MonTraveller app's functional and nonfunctional requirements ● Regularly use the software application and give feedback
Customer Support Staff	They are the first line of contact between end-users and members of the MonTraveller Organization.	● Answer user questions ● Resolve user problems ● Flag bugs reported by users to management
Third-Party Applications	They are software applications not developed by the MonTraveller Organization that are used by the MonTraveller app.	● Process online payments ● Provide maps of Montreal ● Find transportation ● Book accommodations
Montreal Businesses	They are the private businesses located in Montreal that profit from tourism, such as shops, restaurants, hotels, etc.	● Share information, such as news of events and promotions ● Sponsor posts on the application for advertisement
App Store	This is a virtual store where applications can be downloaded on mobile devices.	● Imposes standards and guides on applications in order to include them in the store ● Contributes to app visibility by categorizing it and featuring it in specific lists
Government	The Government of Quebec and Canada constitute Montreal's provincial and federal government respectively.	● Impose regulations on businesses ● Collect taxes from business revenue

3.2. User Environment

The MonTraveller app is a mobile application designed to run on cell phones and tablets operating via iOS and AndroidOS. It is built to be compatible with several third-party software systems in order to provide the target user with many different functionalities all within a single application. Most of the system's features require a working internet connection, therefore the user needs to have access to wifi in order to use the app, or alternatively have substantial cellular data as part of their smartphone plan. With increased marketing strategies, we aim to target Montrealers, residents of Montreal's neighboring suburbs, and Montreal tourists as users of our proposed software solution. As such, we expect the number of simultaneous users to exponentially increase from several thousands to hundreds of thousands, and eventually to surpass the million mark. With this consideration, the MonTraveller app needs to be able to accommodate a growing number of users over time. Though the length of a task cycle depends on multiple factors and of the feature being used, we aim to align the average response time to the industry standard by keeping below 1s. Similarly, the amount of time spent in each activity depends on the user's choice of function. For example, we expect that a user's time spent rating a travel itinerary to be significantly shorter than searching for a restaurant, since the latter activity is dependent on several factors such as budget constraints, dietary restrictions, personal preferences for food and ambiance, business hours and table availability.

4. PRODUCT OVERVIEW

This section presents the flow of data between the main components of the app and explains how said data is used in each of those components. Additionally, it includes the project's current assumptions about the environment it will run in along with their associated dependencies.

4.1. Product Perspective

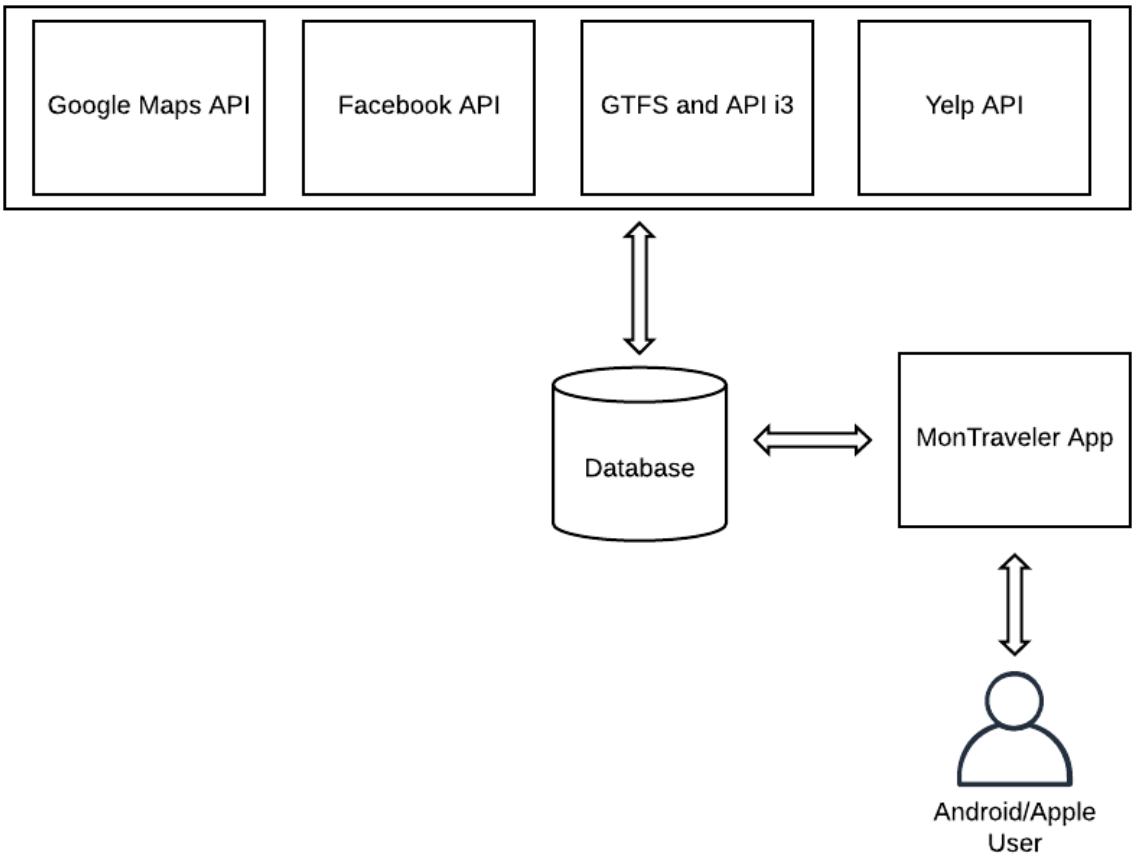

Figure 1: Component Diagram of the MonTraveller app

The user makes requests through the MonTraveller app (requests such as login, sign up) which are then handled by the API servers and checked with the database for validation in the case of a returning user or storage in the case of a new user. Requests involving geo-location are handled by the Google Maps API, while requests involving public transportation are handled by GTFS - realtime data (real-time bus schedules and locations) and API i3 (métro, buses and elevators service updates). When the user wishes to access reviews of certain locations, those are handled by the Yelp API. Requests such as "Share itinerary on Facebook" are handled by Facebook API. The responses from these requests are then sent back to the mobile interface for the user to see.

4.2. Assumptions and Dependencies

Assumptions	Dependencies
The mobile device is running AndroidOS or iOS.	The application solely runs on android or iOS.
Merchants support payment through API requests.	Seamless payment integration requires merchants to provide API endpoints.
The GoogleMaps API is available.	Transportation services used by the application to consume the API for itinerary generation.
Reviews are available online	Reviews provided by the app need to be obtainable.
The Facebook API is available.	Multiple aspects of the app depend on the Facebook API.
The Yelp API is available	The app's review feature depends on the Yelp API.
PCI Compliance	Since credit card numbers are transferred through the app, MonTraveller's infrastructure needs to be regularly checked for PCI compliance.

4.3. Needs and Features

Need	Priority	Features	Planned Release
View map of Montreal	high	Display map	iteration 1
View public transportation maps	high	Display list of transportation options	iteration 1
	high	Display metro map	iteration 1
	high	Display bus lines	iteration 1
	high	Display train lines	iteration 1
Search for places/activities by postal codes	high	Search by postal code	iteration 1
Search for places/activities by the type of activity they wish to do	high	Search by activity	iteration 1
Search for places/activities their budget	normal	Search by budget	iteration 2
Search for places/activities based on ratings on Google maps	normal	Search by rating	iteration 2
Search for places/activities based age restrictions (children, underage minors)	normal	Search by age restrictions	iteration 2
Search for activities/places based on the date and time they will be visiting	high	Search by date/time	iteration 1
View reviews of a selected location/activity	high	Display reviews	iteration 1
Pin the places they wish to visit on to the map	high	Pin to map	iteration 1
Make an itinerary based on pinned search results	high	Make an itinerary	iteration 1
View transportation option based on itinerary	high	Display transportation options	iteration 1

Estimate the time to arrive to the destination	high	Traveling time estimation	iteration 1
Estimate the cost of going on an itinerary	high	Traveling cost estimation	iteration 1
View planned itinerary in calendar	normal	View activity calendar	iteration 2
Share itinerary on Facebook	normal	Share itinerary	iteration 2
Post itinerary to MonTraveller's timeline	normal	Post to timeline	iteration 2
Rate itinerary posted to MonTraveller's timeline (by other users)	normal	Rate itinerary	iteration 2
Leave a review for activity and/or locations contained in itinerary	normal	Write review for activity/location	iteration 2
Find accommodation	high	Display list of popular accommodation	iteration 1
	high	Book accommodations	iteration 1
	high	Make payment for accommodations	iteration 1
Find restaurant	normal	Display list of popular restaurants	iteration 1
	normal	Search by dietary restrictions	iteration 1
	normal	Book restaurant	iteration 1
	normal	Make payment for restaurant booking	iteration 1
Find entertainment/attraction	low	View list of popular entertainment/attraction	iteration 3
Find shops	low	View list of popular shops	iteration 3

4.4. Alternatives and Competition

Competition	Strengths	Weaknesses
Expedia	● It provides many options with flights, such as hotel bookings and vacation promotions. ● Itineraries are available to post to facebook. ● It can save search results and items. ● Has specific page for certain countries	● Recommended activities are limited. Mostly tour guides. ● Activity reviews are on site. ● No local community recommendations
Hotwire	● Provides best deals from unsold rooms. ● Car rental shows car model, seats, deals.	● Won't provide details of the rooms, hotels, cars until after you paid for them.
Trivago	● Shows best deals for hotels	● Limited to display only. It does not book hotels ● Only limited to hotels viewing
Airbnb	● Alternative to hotels ● Wide range of options of room, property ● Provides reviews from users ● Provides pictures of property, rooms	● Only provides room. ● Not a full trip app
Alltrails	● Provides detailed path ● Path have reviews ● Stat recording, including distance traveled, speed, elevation ● GPS feedback on user's position on trail	● Requires to pay to access all contents and features ● For hiking and biking
Culturetrip	● Can be used offline ● Provides reviews for trips, hotels, restaurants ● Can save items to wishlist	
Gasbuddy	● Provides gas stop convenience store locations ● Provides price information	● Mainly for road trips
Google Maps	● Provides different transportation options ● Map zooms to pictures of roads and buildings	

Tripcase	• Books hotels, flights, details into one itinerary	• Lack of activity recommendations • Pro account for full features • Better suited for flights
Viator	• Activity highlights • Daily activity recommendations • User posts of reviews and pictures • Social media sharing	
Yelp	• Finding restaurants on locations • Filters to match criteria • User reviews and pictures	
Zomato	• Find restaurants on regions • Reviews and pictures	
Wanderu	• Book flights, trains, buses for between city travel • Can compare prices • Itineraries of travels	• Inner city travel less detailed • No activity

Most of the competition does well in their exclusive domain of reviewing businesses, finding travel directions, or booking hotels. However, they do not provide the full experience of Montreal on a single app and users need to use multiple software which would require a good amount of organization. In addition, the main competitive tourist applications only list a small subset of activities and restaurants. We are building a system that incorporates pathing, hotels, and activities into one. Some features will be embeddings of other software (Google Maps, Yelp, Facebook) to avoid building features that already exist and work very well. A bigger subset of activities and events will be available to the user on the app as opposed to just tourist events. The idea is to simplify and gather all the best features into one application and provide an enriching experience of Montreal for visitors without the need of using multiple applications.

5. OTHER PRODUCT REQUIREMENTS

This section presents a collection of nonfunctional requirements for the MonTraveller app. These additional product requirements describe how the software system should behave and are categorized under the following properties: applicable software standards, hardware and platform requirements, performance requirements, environmental requirements, robustness, usability, availability, reliability, security, design constraints, and documentation requirements. A description of each property is given in the table below.

Requirements	Priority	Description
Applicable Standards	high	● The system will use the industry standard OAuth2 authentication method for social media integration. ● Communication between application servers and users will be under HTTPS. ● Compliant with ISO International Standards
Hardware or Platform Requirements	high	● User must have a smartphone or tablet with a working internet connection and sufficient storage to download and use the app ● The user's mobile device must be running AndroidOS or iOS
Performance Requirements	high	● The system must have a quick average response time. Despite the number or users and amount of data being processed, users expect mobile apps to have fast performance. User interface events should have a maximum acceptable response delay of 1000ms. ● The system should be able to handle 20 million simultaneous users without loss of performance ● The system should be able to identify users and reopen their sessions
Environmental Requirements	normal	● The application will be energy efficient by being *Lazy First*. [8]
Robustness	high	● The system must minimize mean time between failures ● The system must have a high fault tolerance and continue its normal operation despite faults within the system's components

Usability	high	● Users should easily be able to use and operate the app with visual cues and clear instructions without any prior training. ● Users should be able to complete each activity using only a small number of operations. ● The system shall display clear prompts to the user in order to login, logout, select preferences and submit requests.
Availability	normal	● MonTraveller should be accessible 24/7. The application should be able to run any day any time.
Reliability	normal	● The system should have less than 2 hours of downtime a month for software updates and bug fixes.
Security	high	● The system must identify account users with a valid username and password before allowing them to access any personal information. ● The system shall be equipped with encryption algorithms to protect against security threats (ex: credit card fraud).
Design constraints	high	● Memory usage is constrained on a mobile device, so the MonTraveller app mustn't take too much space on the user's phone or tablet to run.
Documentation requirements	low	● Installation and setup instructions will appear on the App Store. ● The app will have a *Help* tab with frequently asked questions and answers that will act as a user's manual. ● Online help will be provided to users through a messaging feature by Customer Support Staff.

6. USE CASE BRIEFS

Actor	**Goal**
User	Plan Itinerary Book Restaurant Find Entertainment Book Accommodation

UC-1	Plan Itinerary
Actor	User
The user searches for an activity, the system generates results matching the search criteria. The user selects *Add to Itinerary* and repeats the last two steps until they're satisfied. The user configures the itinerary by inputting the activities' start and end date along with the preferred method of transportation between each activity. The user then selects *View Itinerary* and the user is redirected by the system to a generated page showing a map for the itinerary, along with a list of directions to complete the itinerary. After the last activity's completion date, the user is sent an email asking for feedback on the itinerary.	

UC-2	Find Entertainment
Actor	User
The user selects *Things To Do* to find entertainment and attractions in Montreal. The user inputs search parameters by specifying preferences and constraints and submits search. The system displays a list of results. The user selects desired search results to learn more about the event/attraction.	

UC-3	Book Restaurant
Actor	User
The user retrieves desired restaurant from restaurants listing and selects *Make a Reservation*. The system redirects the user to the restaurant's booking service. The user inputs the date, time, and number of people into the booking system and searches for a table. The system displays available tables given the input parameters. The user selects	

desired reservation, enters contact information and submits the restaurant booking. The system displays a confirmation of the booking and adds it to itinerary and calendar.

UC-4	Book Accomodation
Actor	User
The user retrieves desired accommodation from accommodations listing and selects *Make a Reservation*. The system redirects the user to the accommodation's booking service. The user selects check-in and check-out dates, number of guests, and searches for a room. The system displays all available options given input parameters. The user selects the desired option, enters contact and payment information and submits booking. The system displays a confirmation of the booking and adds it to the itinerary and calendar.	

UC-5	Leave Fake Reviews
Actor	Hostile Agents
The hostile agent retrieves activity or location and selects *Leave a Review*. The system displays the rating and review page. The hostile agent writes a fake positive review to increase the business's revenue or writes a fake negative review to attack the business and sink their ranking. The hostile agent submits the review. The system displays the review and rating along with the reviewer’s username on the activity’s or location’s page.	

UC-6	Spam Search Requests
Actor	Hostile Agents
The hostile agent completes multiple search queries in order to overwhelm the system’s resources, rendering it unable to serve legitimate users.	

7. FULLY DRESSED USE CASES

Id: UC- 3.0

Use Case: Book a Restaurant

Description
The user inputs their constraints in the *Book a Restaurant* page. After looking through the curated list generated by the system, the user selects their preferred location. Once user credentials are inputted, the system makes a reservation and a confirmation email is sent to the user's inbox.

Level: User Goal

Primary Actor: User

Supporting Actors: Restaurant Booking System, Review Provider

Stakeholders and Interests
User: Wishes to book a restaurant that suits their needs.
Restaurant Booking System: Wishes to make the booking process as simple as possible.
Review Provider: Wishes to reach and help as many customers as possible.

Pre-Conditions
1. User is authenticated by the system

Post Conditions
Success end condition
1. The restaurant is booked at specified time and location
2. Confirmation email is sent to user's inbox

Failure end condition:
1. Failure to book alert is displayed

Minimal Guarantee
1. Confirmation email is sent to user's inbox

Main Success Scenario
1. The user selects *Book a Restaurant*
2. The user inputs postal code
3. The user inputs check-in and check-out dates
4. The user selects desired criteria (rating, budget, distance, dietary restrictions)
5. The user submits search
6. The system displays a list of restaurants fitting user's criteria
7. The user selects a restaurant from the list
8. The system prompts the user to enter personal and contact information

9. The user inputs the personal and contact information
10. The restaurant is booked for the user

Extensions

2a. Entered postal code is invalid, indicate error.
6a. There are no matches for the user's needs, indicate error.
7a. The user wishes to consult reviews.
9a. The inputted user information are invalid, indicate error.

Special Requirements

The user must be able to navigate the application and input information through a touchscreen UI.

Id: UC- 4.0

Use Case: Book Accommodation

Description:
The user chooses an accommodation from the accommodation listing and selects *Make a reservation*. The system displays the accommodation booking service. The user proceeds to select the dates for checking in and checking out the number of guests. The system displays the available rooms for the selected criteria. The user selects the desired rooms to book. The user selects the payment options, contact and payment information and submits. The system displays a confirmation of the booking and the booking is added to the user's itinerary and calendar.

Level: User Goal

Primary Actor: User

Supporting Actors: Accommodation Booking System, Review Provider

Stakeholders and Interests
User: Wants to find an accommodation suitable to the user's needs.
Accommodation Booking System: Wants to have as many bookings without user complaints
Review Provider: Wants to reach and help as many customers as possible
Pre-Conditions
User is authenticated by the system

Post Conditions:
Success end condition:
1. The selected accommodation is booked for the selected dates.
2. The payment is sent.
3. A confirmation is displayed.
4. The system sends a confirmation email to the user.

Failure end condition:
1. The payment is canceled.
2. An unsuccessful confirmation is displayed.

Minimal Guarantee
1. A confirmation of booking status is displayed.
2. A confirmation email is sent to the user.

Main Success Scenario
1. The user selects *choose accommodation*
2. The user inputs their postal code
3. The user inputs check-in and check-out dates
4. The user selects and submits desired criteria (rating, budget, distance, room size, features)
5. The system displays a list of accommodations fitting the user's criteria
6. The user selects an accommodation (ex: a hotel room) from the list
7. The system prompts the user to enter personal and contact information
8. The user inputs and submits their personal and contact information

9. The system prompts payment options
10. The user selects the desired payment option and pays
11. The system processes the payment
12. The accommodation is booked for the user

Extensions

2a. The inputted postal code is invalid and invalid message is displayed
4a. The inputted criteria results in 0 options available
5a. The user reads the review given by the Review Provider
8a. The inputted user information are invalid and invalid message is displayed
11a. Payment fails and payment error message is displayed

Special Requirements :

The user must be able to navigate the application and input information through a touchscreen UI.

8. USE CASE DIAGRAM

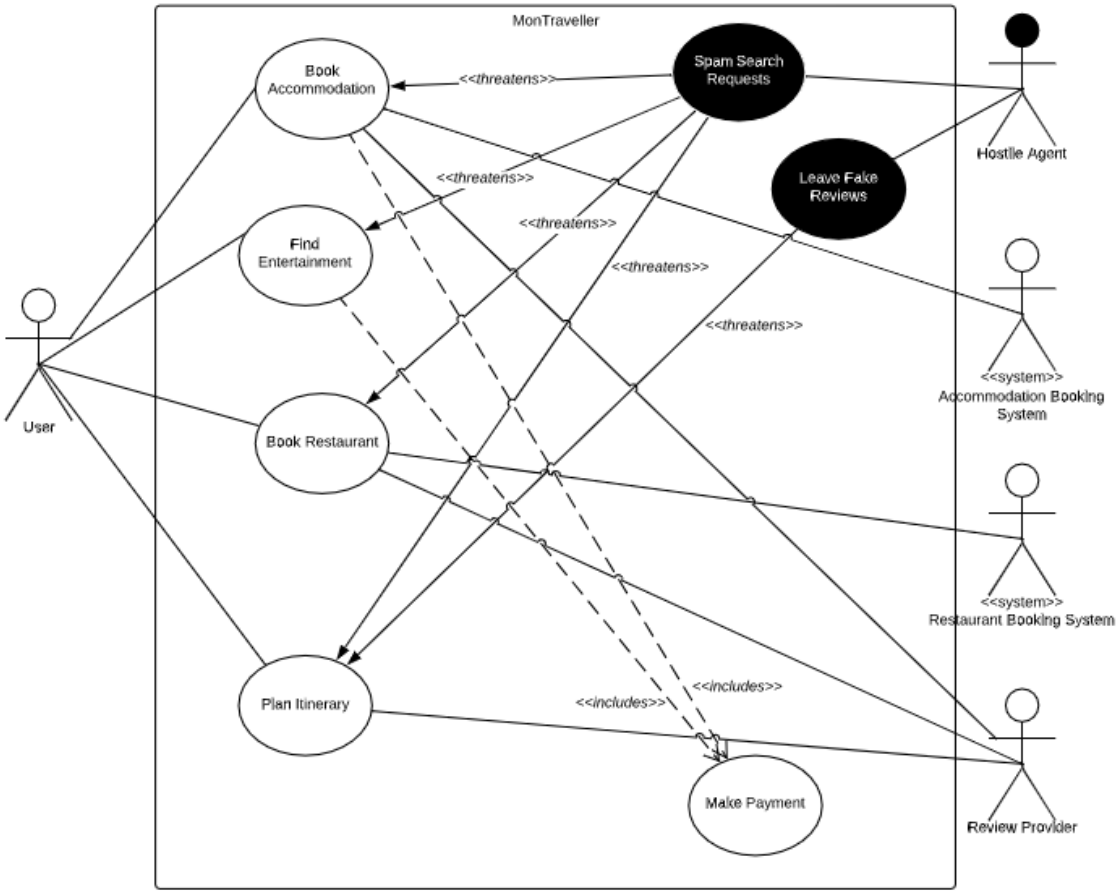

Figure 2: Use Case Diagram of the MonTraveller app

9. SUPPLEMENTARY SPECIFICATION

9.1. Introduction

9.1.1. Purpose

The purpose of the Supplementary Specification document is to capture those system requirements of the Montraveller application that cannot be readily expressed within the use-case model. These requirements are categorized into three classes: functional requirements, non-functional requirements, and design constraints. Together, MonTraveller's use case model and Supplementary Specifications provide a full description of the requirements placed on the system to be built.

9.1.2. Scope

This Supplementary Specification applies to the MonTraveller system introduced in the Vision Document. The MonTraveller application will enable Montrealers, residents of Montreal's suburbs, and tourists to Montreal to find up-to-date information about the goings-on in the city. Users will be able to create and share itineraries for trips in the city, find and book accommodations and entertainment, as well as browse restaurants lists and make dinner reservations. In this document, we record a set of additional requirements for the MonTraveller application that play an important role in defining system behaviors which may not be as externally visible as those presented earlier in the form of use cases.

9.1.3. Definitions, Acronyms and Abbreviations

Refer to the Glossary in section 14 of the Supplementary Specification.

9.1.4. Overview

In the following sections of the report, we provide a list of supplemental requirements of the MonTraveller app in order to ensure comprehensiveness and completeness of the software

solution's specifications. In section 2, we list functional requirements that have not been captured by use cases. Sections 3 through 6 present different categories of non-functional requirements. In section 3, we specify requirements affecting the mobile application's usability. Section 4 outlines those requirements that influence both the system's reliability and availability. Section 5 identifies system characteristics that define the performance desired by the software developers and expected by the end-users. In section 6, we define the system's supportability by enumerating those requirements that will enhance its ability to accommodate additional features in the future. Section 7 describes the limitations which are imposed on the design and development of the MonTraveller app by listing design constraints. In Section 8, we explain the different types of online user documentation and help system requirements for the MonTraveller application. Section 9 describes purchased components to be used within the system. Section 10 defines user, hardware, software, and communication interfaces that must be supported by the application. Section 11 includes licensing and security requirements to be implemented. Section 12 lists any legal disclaimers or copyright notices as they pertain to our system. Section 13 notes the sections of applicable standards of software development that apply to the MonTraveller app. Lastly, section 14 concludes the Supplementary Specification with a glossary of terms and acronyms.

9.2. Functionality

9.2.1. Plan an Itinerary

This feature allows the user to plan an itinerary by giving access to maps and reviews.

1. The system shall allow the user to view maps.
2. The system shall allow the user to view metro maps.
3. The system shall allow the user to view train lines and itineraries.
4. The system shall allow the user to view bus lines and itineraries.
5. The system shall allow the user to view a list of transportation options.
6. The system shall allow the user to pin items on the map.
7. The system shall allow the user to make an itinerary.
8. The system shall allow the user to view their activity calendar.
9. The system shall allow the user to view a generated time estimate for a given itinerary.
10. The system shall allow the user to view a generated cost estimate for a given itinerary.
11. The system shall allow the user to share their itinerary to Facebook.
12. The system shall allow the user to post their itinerary to their timeline.
13. The system shall allow the user to rate another user's itinerary.

9.2.2. Find Entertainment

This feature allows the user to find suitable entertainment options by giving access to reviews and search ability to search according to aspects that are of importance to the user.

1. The system shall allow the user to search by postal code.
2. The system shall allow the user to search by activity.
3. The system shall allow the user to search by budget.
4. The system shall allow the user to search by rating.
5. The system shall allow the user to search by age restriction.
6. The system shall allow the user to search by date and or time.
7. The system shall allow the user to view a list of popular attractions and entertainment.
8. The system shall allow the user to view a list of popular shops.
9. The system shall allow the user to leave a review for an activity or a location.
10. The system shall allow the user to view reviews for an activity or a location.
11. The system shall allow the user to make payments.
12. The system shall allow the user to pin items on the map.

9.2.3. Book Accomodation

This feature allows the user to book suitable accommodation options by giving access to reviews and ability to search according to aspects that are of importance to the user.

1. The system shall allow the user to search by postal code.
2. The system shall allow the user to search by budget.
3. The system shall allow the user to search by rating.
4. The system shall allow the user to search by age restriction.
5. The system shall allow the user to search by date and or time.
6. The system shall allow the user to view a list of popular accommodations.
7. The system shall allow the user to leave a review for the accommodation.
8. The system shall allow the user to view reviews for the accommodation.
9. The system shall allow the user to make payments.

10. The system shall allow the user to pin items on the map.

9.2.4. Book Restaurant

This feature allows the user to book suitable restaurant options by giving access to reviews and ability to search according to aspects that are of importance to the user.

1. The system shall allow the user to search by postal code.
2. The system shall allow the user to search by budget.
3. The system shall allow the user to search by rating.
4. The system shall allow the user to search by age restriction.
5. The system shall allow the user to search by date and or time.
6. The system shall allow the user to search by dietary restrictions.
7. The system shall allow the user to view a list of popular restaurants.
8. The system shall allow the user to leave a review for the restaurant.
9. The system shall allow the user to view reviews for the restaurant.
10. The system shall allow the user to book a restaurant.
11. The system shall allow the user to pin items on the map.

9.3. Usability

9.3.1. Training Time - Normal User

The training time for a normal user to plan an itinerary shall be under 45 minutes.

9.3.2. Training Time - Power User

The training time for a power user to plan an itinerary shall be under 20 minutes.

9.3.3. Back Button Support

Pressing the back button shall not invalidate a logged-in user's session.

9.3.4. Colour Blindness

The application shall avoid colors associated with color blindness for map generation.

9.3.5. Itinerary Generation

The time for a normal user to generate an itinerary consisting of 5 components (restaurants, etc.) shall be under 45 minutes.

9.3.6. User Manual

The user manual shall be accessible in every page of the application via a question mark in the top right corner. The user manual shall include step-by-step instructions on how to use the system features. The user manual shall include a glossary of definitions, acronyms, and abbreviations.

9.4. Reliability

Weekly Maintenance

The system shall have a maximum of 15 minutes of downtime for maintenance per week.

Mean Time To Repair

The system should be down for at most 10 minutes after failure.

Maximum Minor Defect Density

The maximum number of minor, significant and critical bugs/KLOC shall not exceed 5, 10 and 20 respectively.
A critical bug results in loss of data, a significant bug results hinders the performance of a system service and a minor bug produces errors affecting long term system maintenance.

Mean Time Between Failures

The mean time between failures shall not exceed an hour.

9.5. Performance

9.5.1. User Interface Response Time

The application must have a quick response time for the user. The input delay between the user and the UI should be under 10ms.

9.5.2. Process Payment Response Time

The response time for the payment to process should not exceed over 5 seconds with an average of 3 seconds per process. This relates to UC2, UC3, and UC4.

9.5.3. Transactions per Second

The application should be able to handle 1000 transactions per second during peak times.

9.5.4. Generate List time

Listings of bookings and entertainment should be generated on average under 1 second for every list of 1000 options, with a maximum time of 3 seconds. This relates to UC2, UC3, and UC4.

9.5.5. Application Memory Size

The overall application should be under 100 MB in memory size.

9.5.6. Increase User Size Performance

The application's performance should not degrade when there are increase in users and data being processed. The application should be capable of handling 100 000 users at the same time.

9.5.7. Response Time Delays due to Server Communication Problems

If server response time exceeds 10 seconds, the connection of the user application and the server will be deemed non-acceptable and will cancel all data transfers and display a communication error.

9.6. Supportability

9.6.1. Coding Standards

The application will be written following the MISRA standards to ensure safe, secure, and reliable code.

9.6.2. Naming Conventions

All variables, keywords will be in lowercase and will use underscores (_) to separate words. Macros will be in full uppercase and its words will be separated by underscores.

9.6.3. Class Libraries

All class libraries must be up to date for every weekly maintenance.

9.6.4. Maintenance Utilities

Maintenance utilities will be used to monitor impact analysis of changes in the system, such as added features on the updates.

9.7. Design Constraints

9.7.1. Supported Platforms

The application shall be able to be run on iOS, and Android.

9.7.2. Distributed Systems

The backend components of the system shall be run on Kubernetes clusters.

9.7.3. Backend Programming Language

The programming language of choice to develop the backend components of the application shall be one of, or a combination of the following: Java, C#, Python, JavaScript.

9.7.4. Frontend Programming Language

The frontend programming language for this application shall be in Java for Android and Swift for iOS.

9.8. Online User Documentation and Help System Requirements

9.8.1. User documentation

User documentation shall be separated into the following sections:

- Booking a Restaurant on MonTraveller
- Booking an accommodation on MonTraveller
- Using MonTraveller to plan an itinerary
- Using MonTraveller to Find Entertainment

9.8.2. Video Tutorials

Every user documentation section shall include a video tutorial explaining the purpose of this section of the app for the user.

9.8.3. Email Support

A support email shall be set up to respond to user requests pertaining to platform usage.

9.9. Purchased Components

n/a

9.10. Interfaces

9.10.1. User Interfaces

The application will utilise a fully touchscreen interface designed for android and iOS systems. All user inputs will be entered through the touchscreen.

The UI will support a simple intuitive navigation with touch. For example, when the application displays the list of booking options of UC2, UC3, and UC4, the user can navigate through them by simply dragging a finger through the list. Choosing an option will be done by a simple touch of the desired selection.

9.10.2. Hardware Interfaces

All data information will be saved in a remote server within Montreal that can be connected by any wireless device with the appropriate security logs. The server will hold 500 terabytes of data.

The user application will only be supported by smartphones with touchscreens. There are no other navigation controls besides the touchscreen that will be supported.

9.10.3. Software Interfaces

Interactive interfaces and embedded systems to support the MonTraveller application include

- Paypal and credit card support for payments
- Google Map API for transportation, maps, directions, etc.
- Facebook API for social media
- Yelp API for reviews

9.10.4. Communications Interfaces

All devices with the MonTraveller application installed need to establish a stable connection to the internet to be able to fully utilize the features. Unstable connections or offline status will result with the application displaying a connection error to the server. The user interface will prompt the user to go to the user's device settings and to establish a connection before proceeding with the application.

9.11. Licensing Requirements

n/a

9.12. Legal, Copyright and Other Notices

n/a

9.13. Applicable Standards

9.13.1. PIPEDA

The application shall adhere to PIPEDA regarding customer data storage and sharing with third-parties.

9.13.2. GDPR

The application shall adhere to GDPR regarding customer data storage and sharing with third-parties.

9.13.3. PCI DSS Compliance

The application shall follow the guidelines set forth by the PCI Security Standards Council.

9.13.4. ISO 9241

The application shall adhere to the ISO 9241 standard for evaluation of application software quality.

9.13.5. ISO 9126

The application shall adhere to the ISO 9126 standard for evaluating a software application's external qualities such as usability, reliability, etc.

10. GLOSSARY

Term	Definition
AndroidOS	The operating system running on Android devices.
API	Application Programming Interface
Assumption	The state of an act assumed to be true and verified at a later time.
Environment	The hardware and software used by the system.
Embedding	An integration of a system into another system.
Features	Functional requirements that satisfy needs.
GTFS	General Transit Feed Specification defines a common format for public transportation schedules. [6]
Interface	Device or a system that unrelated entities use to interact [7]
Iteration	Development steps of the project, where each step is an improvement over the previous one.
iOS	The operating system running on Apple devices.
Itinerary	A documented travel path
Montrealer	Person that resides in Montreal.
Needs	A want or necessity that the system fulfills.
Performance	The measurability of an action of the system.
Server	A server is a computer designed to process requests and deliver data to another computer over the internet or a local network. [5]
Stakeholder	An individual or an organization that will be affected (positively or negatively) by the implementation of this project.
User	Person or software that employs the system.

11. REFERENCES

1. Axel van Lamsweerde: *Requirements Engineering: From System Goals to UML Models to Software Specifications.* Wiley, 2009.
2. Dean Leffingwell, *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise.* 1st ed., Addison-Wesley Professional, 2011.
3. Dean Leffingwell, Don Widrig: *Managing Software Requirements: A Use Case Approach*. 2nd ed., Addison-Wesley, 2003.
4. Larman, C.: *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development* (3rd Edition). Prentice Hall PTR (2004).
5. Mitchell, B. (2019). *What is a Server?*. [online] Lifewire. Available at: https://www.lifewire.com/servers-in-computer-networking-817380 [Accessed 10 Oct. 2019].
6. "GTFS Static Overview | Static Transit | Google Developers." *Google*, Google, https://developers.google.com/transit/gtfs.
7. "What is an Interface?", *The Java Tutorial*. [Online]. Available: https://www.iitk.ac.in/esc101/05Aug/tutorial/java/concepts/interface.html. [Accessed: 10-Oct- 2019].
8. "Optimize for battery life" *Google*, Google, https://developer.android.com/topic/performance/power.