



CSC 431

Wanderlust Travel Recommender

Software Requirements Specification (SRS)

Team 7

Cecile Darwiche	Scrum Master
Michael Sampino	System Architect
Sydney Divozzo	Requirements Engineer

Version History

Version	Date	Author(s)	Change Comments
1.0.0	2/21/24	Cecile Darwiche, Sydney Divozzo, Michael Sampino	First Draft
1.1.0	5/1/24	Cecile Darwiche, Sydney Divozzo, Michael Sampino	Final Draft - Corrected all comments from TA

Table of Contents

1. System Requirements	8
1.1. Functional Requirements	8
1.1.1. Create Account	8
1.1.2. Validate Login	8
1.1.3. Reset Password	8
1.1.4. Change Password	9
1.1.5. Modify Account Settings	9
1.1.6. Read User Preferences	9
1.1.7. Return Recommendations	10
1.1.8. Submit User Reviews	10
1.2. Non-Functional Requirements	11
1.2.1. Password Encryption and Security	11
1.2.2. Automated Password Recovery Email	11
1.2.3. Uptime	11
1.2.4. Interference	11
1.2.5. Model Replacement Downtime	12
1.2.6. Concurrent Users	12
2. System Constraints	13
2.1. Tool Constraints	13
2.1.1. Development Tools	13
2.1.2. Database Management System	13
2.1.3. Version Control System	13
2.1.4. Frontend Technologies	13
2.2. Language Constraints	13
2.2.1. Frontend Development	13
2.2.2. Server-Side Development	14
2.2.3. Database Query Language	14
2.3. Platform Constraints	14
2.3.1. Web Browsers Compatibility	14
2.3.2. Device Compatibility	14
2.4. Hardware Constraints	15
2.4.1. Minimum System Requirements	15
2.4.2. Minimum Processor Speed	15
2.5. Network Constraints	16
2.5.1. Internet Connectivity	16
2.5.2. Bandwidth Requirements	16

2.5.3. Database Connectivity	16
2.6. Deployment Constraints	16
2.6.1. Limited Maintenance Windows	16
2.6.2. Dependency on Third-Party APIs	16
2.7. Transition & Support Constraints	17
2.7.1. Neural Network Maintenance Concern	17
2.7.2. End of Life	17
2.8. Budget & Schedule Constraints	17
2.8.1. Budget Constraint	17
2.8.2. Semester End Constraint	17
2.9. Miscellaneous Constraints	18
2.9.1. Data Backup and Disaster Recovery Constraint	18
3. Requirements Modeling	19
3.1.1. System Use-Case Diagram	19
4. Evolutionary Requirements	20
4.1. Functional Requirements	20
4.1.1. Accommodation Bookings	20
4.1.2. Flight Tracking	20
4.2. Non-Functional Requirements	20
4.2.1. Accessibility	20
4.2.2. Internalization and Localization	20

Table of Tables

	7
1. System Requirements	8
1.1. Functional Requirements	8
Table 1: Create Account	8
Table 2: Validate Login	8
Table 3: Reset Password	8
Table 4: Change Password	9
Table 5: Modify Account Settings	9
Table 6: Read User Preferences	9
Table 7: Return Recommendations	10
Table 8: Submit User Reviews	10
1.2. Non-Functional Requirements	11
Table 9: Password Encryption and Security	11
Table 10: Automated Password Recovery Email	11
Table 11: Uptime	11
Table 12: Interference	11
Table 13: Model Replacement Downtime	12
Table 14: Concurrent Users	12
2. System Constraints	13
2.1. Tool Constraints	13
Table 15: Development Tools	13
Table 16: Database Management System	13
Table 17: Version Control System	13
Table 18: Frontend Technologies	13
2.2. Language Constraints	13
Table 19: Frontend Development	13
Table 20: Server-Side Development	14
Table 21: Database Query Language	14
2.3. Platform Constraints	14
Table 22: Web Browsers Compatibility	14
Table 23: Device Compatibility	14
2.4. Hardware Constraints	15
Table 24: Minimum System Requirements	15
Table 25: Minimum Processor Speed	15
2.5. Network Constraints	16
Table 26: Internet Connectivity	16
Table 27: Bandwidth Requirements	16
Table 28: Database Connectivity	16
2.6. Deployment Constraints	16

Table 29: Limited Maintenance Windows	16
Table 30: Dependency on Third-Party APIs	16
2.7. Transition & Support Constraints	17
Table 31: Neural Network Maintenance Concern	17
Table 32: End of Life	17
2.8. Budget & Schedule Constraints	17
2.8.1. Budget Constraint	17
Table 33: Semester End Constraint	17
2.9. Miscellaneous Constraints	18
Table 34: Data Backup and Disaster Recovery Constraint	18
4. Evolutionary Requirements	20
4.1. Functional Requirements	20
Table 35: Accommodation Bookings	20
Table 36: Flight Tracking	20
4.2. Non-Functional Requirements	20
Table 37: Accessibility	20
Table 38: Internalization and Localization	20

Table of Figures

3. Requirements Modeling	19
Figure 1: System Use-Case Diagram	19

1. System Requirements

1.1. Functional Requirements

1.1.1. Create Account

Table 1: Create Account

Title	Create Account
Description	Form where user inputs their personal information that will be associated with their account
Priority	0
Precondition(s)	Navigate to the website and click "Create Account"
Basic Flow	After clicking "Create Account", user is prompted to input their name, age, gender, email and password After inputting this information, an account is created for the user and this information is stored in a database with an encrypted password.
Postconditions(s)	User is taken to the login screen
Use Case Diagram	3.1.1

1.1.2. Validate Login

Table 2: Validate Login

Title	Account Login
Description	The user is prompted to input the email and password associated with their account
Priority	0
Precondition(s)	User must create an account
Basic Flow	After the user creates an account, they are prompted to enter the username and password associated with the account The user presses "Sign In" and the system validates their account in the database
Postconditions(s)	If the account exists, then user is taken to the homepage of the website If the account is not found, the user is prompted to try logging in again, create an account or reset password
Use Case Diagram	3.1.1

1.1.3. Reset Password

Table 3: Reset Password

Title	Reset password
Description	The user is prompted to input the email associated with their account and they will be emailed a new randomized password.

Priority	0
Precondition(s)	User must have an existing account and input the email associated with the account
Basic Flow	On the login page, user will click "forget password" and they will be redirected to a page where they must input their email
Postconditions(s)	If the account exists, the user will be sent an email with a randomized new password that they can change in their settings upon a successful login.
Use Case Diagram	3.1.1

1.1.4. Change Password

Table 4: Change Password

Title	Change password
Description	The user can change their password upon successful login in the account settings page.
Priority	3
Precondition(s)	User must be logged in and know their existing password
Basic Flow	Users must navigate to the settings page. On this page, they will have an option to select "change password". Then will then be redirected to a page where they must input their current password, and then the new password that they want twice.
Postconditions(s)	The page will display a message stating that the password has been changed if the old password was correct and the 2 new password fields match.
Use Case Diagram	3.1.1

1.1.5. Modify Account Settings

Table 5: Modify Account Settings

Title	Modify Account Settings
Description	Allow users to change their account settings
Priority	3
Precondition(s)	User must be logged in
Basic Flow	User must click on "Settings" from the navigation bar User is taken to a separate "Settings" page in which they can change their name, age, gender, email User can also chose not to share their activity reviews on the Wanderlust website After modifying their settings, the user is asked if they want to save or discard changes
Postconditions(s)	Changes are saved and the user is redirected to the homepage
Use Case Diagram	3.1.1

1.1.6. Read User Preferences

Table 6: Read User Preferences

Title	Read User Preferences
Description	Form where user inputs their upcoming travel plans
Priority	0
Precondition(s)	User must be logged in and click on "Plan my trip" from the navigation bar
Basic Flow	<p>The user is prompted to enter where they are traveling to</p> <p>The user is prompted to enter when they are traveling to said destination</p> <p>The user is prompted to pick which type of activities are most appealing to them from the list provided</p> <p>The user is prompted to describe the occasion for the trip</p> <p>The user is prompted to detail who they are traveling with</p> <p>The user is prompted to enter any dietary restrictions they have</p> <p>The user is prompted to select the types of cuisine they like from the list provided</p> <p>The user submits the form by pressing "Return my recommendations"</p>
Postconditions(s)	Application stores user preferences
Use Case Diagram	3.1.1

1.1.7. Return Recommendations

Table 7: Return Recommendations

Title	Return Recommendations
Description	Based on their input from the form, Wanderlust's AI returns personalized activity recommendations.
Priority	0
Precondition(s)	User must be logged in and must fill out the user preference form
Basic Flow	<p>AI takes the user's information provided from the form(1.1.6) as well as their personal information provided when they created their account to create a list of activities in the city in which the user is staying. These activities will be catered to the user's preference, and will also include restaurant recommendations.</p> <p>Users are also able to view reviews from other users for each individual activity or restaurant they were recommended.</p>
Postconditions(s)	Recommendations are returned to user.
Use Case Diagram	3.1.1

1.1.8. Submit User Reviews

Table 8: Submit User Reviews

Title	Submit User Reviews
Description	Users will be able to submit reviews for activities and restaurants in the different cities they have visited. These reviews can be viewed by other users
Priority	1

Precondition(s)	User must be logged in and click on "Write a Review" from navigation bar
Basic Flow	User will click on "Write a Review" User will search for activity or restaurant they want to review User will be prompted to rate the activity or restaurant from 0-5 stars User will be prompted to enter any comments they would like to add User is asked to submit their review
Postconditions(s)	Review is stored and displayed on the activity or restaurant page
Use Case Diagram	3.1.1

1.2. Non-Functional Requirements

1.2.1. Password Encryption and Security

Table 9: Password Encryption and Security

Title	Password Encryption
Description	The user's password they created upon making an account will be stored as an encrypted password in a database.
Priority	1
Applicable FR(s)	1.1.1

1.2.2. Automated Password Recovery Email

Table 10: Automated Password Recovery Email

Title	Automated Password Recovery Email
Description	When a user selects "Forgot Password" on the login page, an automatic email will be sent with a link to reset their password.
Priority	1
Applicable FR(s)	1.1.2

1.2.3. Uptime

Table 11: Uptime

Title	Uptime
Description	The application will be down for no longer than 20 minutes every month
Priority	4
Applicable FR(s)	N/A

1.2.4. Interference

Table 12: Interference

Title	Interference
Description	Interference for verification must take less than 10 seconds
Priority	4
Applicable FR(s)	1.1.4, 1.1.5

1.2.5. Model Replacement Downtime

Table 13: Model Replacement Downtime

Title	Model replacement downtime
Description	When the parameters are updated by developers, the website should be down for no more than 10 minutes.
Priority	4
Applicable FR(s)	N/A

1.2.6. Concurrent Users

Table 14: Concurrent Users

Title	Concurrent Users
Description	The system will be able to handle a large number of concurrent users without experiencing performance degradation.
Priority	0
Applicable FR(s)	N/A

2. System Constraints

2.1. Tool Constraints

2.1.1. Development Tools

Table 15: Development Tool Constraint

Title	Development Tool Constraint
Description	The project shall exclusively use NodeJS for server-side development
Reference	nodejs.org
Priority	Mandatory: 0

2.1.2. Database Management System

Table 16: Database Management System Constraint

Title	Database Management System Constraint
Description	PostgreSQL shall be the designated relational database management system for the Wanderlust Travel Recommender project.
Reference	postgresql.org
Priority	Mandatory: 0

2.1.3. Version Control System

Table 17: Version Control System Constraint

Title	Version Control System Constraint
Description	Git shall be employed for version control throughout the project, and the codebase will be hosted on GitHub.
Reference	git-scm.com , github.com
Priority	Mandatory: 0

2.1.4. Frontend Technologies

Table 18: Frontend Technologies Constraint

Title	Frontend Technologies Constraint
Description	The client-side development shall be implemented using HTML, CSS, and JavaScript. Compatibility with modern web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge is required.
Priority	Mandatory: 0

2.2. Language Constraints

2.2.1. Frontend Development

Table 19: Frontend Development Constraint

Title	Frontend Development Constraint
Description	The frontend user interface of the Wanderlust Travel Recommender web application shall be developed using HTML, CSS, and JavaScript.
Priority	Mandatory: 0

2.2.2. Server-Side Development

Table 20: Server-Side Development Constraint

Title	Server-Side Development Constraint
Description	The backend of the Wanderlust Travel Recommender shall be developed using Node Js for server-side programming.
Reference	nodejs.org
Priority	Mandatory: 0

2.2.3. Database Query Language

Table 21: Database Query Language Constraint

Title	Database Query Language Constraint
Description	PostgreSQL's SQL shall be utilized for querying the relational database in the backend of the Wanderlust Travel Recommender.
Reference	postgresql.org
Priority	Mandatory: 0

2.3. Platform Constraints

2.3.1. Web Browsers Compatibility

Table 22: Web Browsers Compatibility Constraint

Title	Web Browsers Compatibility Constraint
Description	The Wanderlust Travel Recommender web application shall be compatible with modern web browsers including Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.
Priority	Mandatory: 0

2.3.2. Device Compatibility

Table 23: Device Compatibility Constraint

Title	Device Compatibility Constraint
Description	The Wanderlust Travel Recommender shall be accessible on desktop computers, laptops, and smartphones with internet connectivity.
Priority	Mandatory: 0

2.4. Hardware Constraints

2.4.1. Minimum System Requirements

Table 24: Minimum System Requirements Constraint

Title	Minimum System Requirements Constraint
Description	Users' devices accessing the Wanderlust Travel Recommender should meet minimum system requirements, including a compatible web browser and internet connectivity.
Priority	Mandatory: 0

2.4.2. Minimum Processor Speed

Table 25: Minimum Processor Speed Constraint

Title	Minimum Processor Speed Constraint
Description	The Wanderlust Travel Recommender web application requires a minimum processor speed to ensure basic functionality. Devices used to access the application must have at least 1.6 GHz, which is necessary to handle the computational load without performance degradation.
Priority	Recommended: 3

2.5. Network Constraints

2.5.1. Internet Connectivity

Table 26: Internet Connectivity Constraint

Title	Internet Connectivity Constraint
Description	The Wanderlust Travel Recommender requires a stable internet connection for users to access and interact with the application.
Priority	Recommended: 2

2.5.2. Bandwidth Requirements

Table 27: Bandwidth Requirements Constraint

Title	Bandwidth Requirements Constraint
Description	Users should have sufficient bandwidth to ensure a seamless experience while interacting with the Wanderlust Travel Recommender, especially when accessing multimedia content or making API calls.
Priority	Recommended: 2

2.5.3. Database Connectivity

Table 28: Database Connectivity Constraint

Title	Database Connectivity Constraint
Description	The Wanderlust Travel Recommender, through the server-side implementation, shall establish a secure connection to the PostgreSQL database for storing and retrieving user data.
Priority	High: 1

2.6. Deployment Constraints

2.6.1. Limited Maintenance Windows

Table 29: Limited Maintenance Windows

Title	Limited Maintenance Windows
Description	Deployment may be constrained by limited maintenance windows or downtime allowances for deploying updates or new features.
Priority	Medium: 3

2.6.2. Dependency on Third-Party APIs

Table 29: Dependency on Thirs-Part APIs

Title	Dependency on Third-Party APIs
Description	The system relies on integration with external APIs for functionalities such activity recommendations. Any downtime, changes in API endpoints, or rate limits imposed by these

	third-party APIs could impact system functionality and user experience.
Priority	High: 2

2.7. Transition & Support Constraints

2.7.1. Neural Network Maintenance Concern

Table 30: Neural Network Maintenance Concern

Title	Neural Network Maintenance Concern
Description	If any neural network shows undesirable classification behavior, developers should be able to upload new set of parameters without significant downtime.
Priority	High: 2

2.7.2. End of Life

Table 31: End of Life

Title	End of Life
Description	The developers of this project (unless they fail this class) will cease working on the project at the end of the semester, at which point the project must be retired unless a new team can take over development.
Priority	Lowest: 5

2.8. Budget & Schedule Constraints

2.8.1. Budget Constraint

Table 32: Budget Constraint

Title	Budget Constraint
Description	There is no funding for this project so no budget is necessary.
Priority	Lowest: 5

2.8.2. Semester End Constraint

Table 33: Semester End Constraint

Title	Semester End Constraint
Description	This project is for CSC 431 course and should therefore be finished before the Spring 2023 semester is over.
Priority	Mandatory: 5

2.9. Miscellaneous Constraints

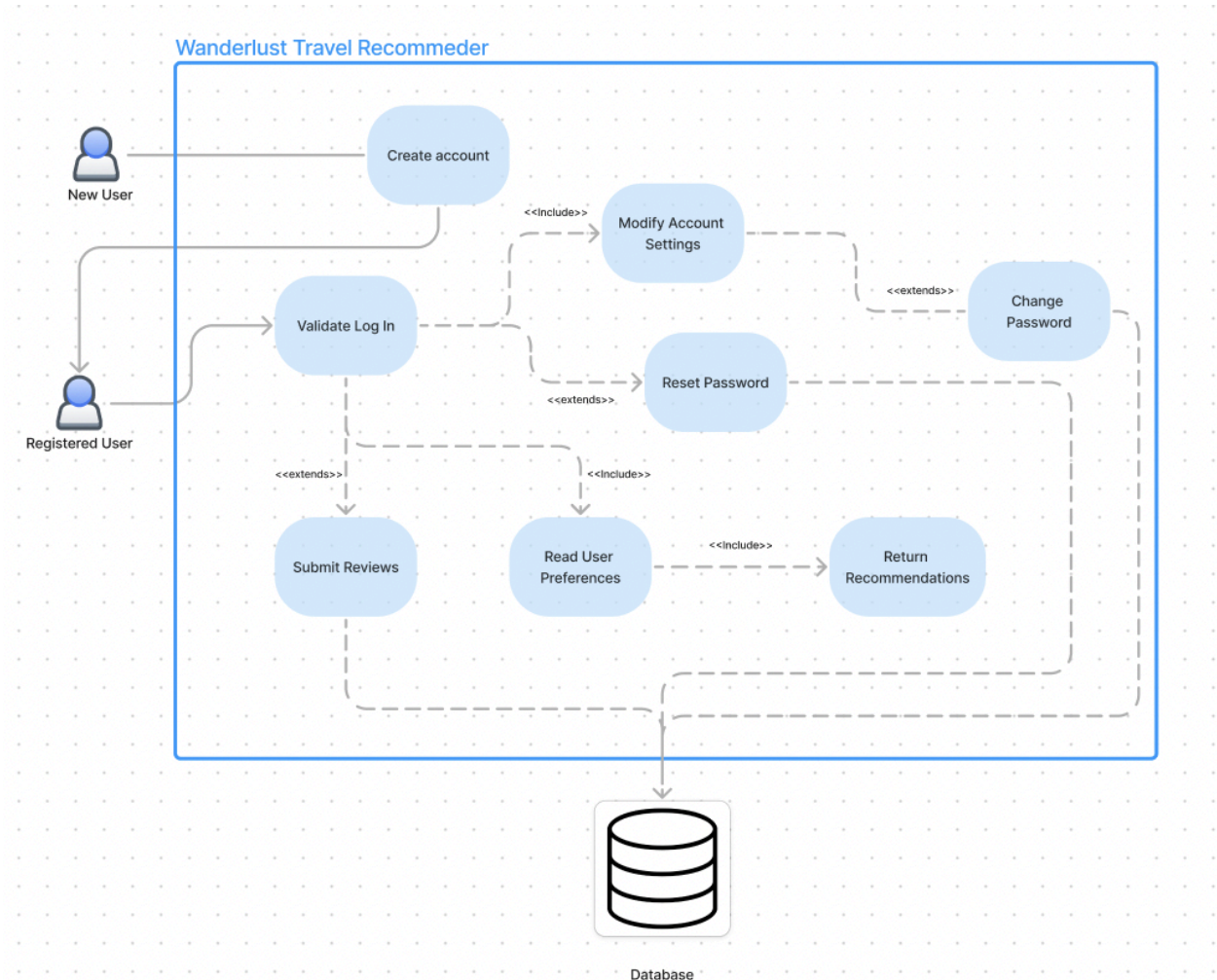
2.9.1. Data Backup and Disaster Recovery Constraint

Table 34: Data Backup and Disaster Recovery Constraint

Title	Data Backup and Disaster Recovery
Description	Implementing robust data backup and disaster recovery mechanisms to protect against data loss, corruption, or system failures.
Priority	High: 2

3. Requirements Modeling

3.1.1. System Use-Case Diagram



4. Evolutionary Requirements

4.1. Functional Requirements

4.1.1. Accommodation Bookings

Table 35: Accommodation Bookings

Title	Accommodation Bookings
Description	The user will have the ability to see accommodation recommendations and book directly through the site.
Priority	5
Precondition(s)	Users must fill out the user preference form
Postconditions(s)	User is able to both plan activities and book hotels on the website
Use Case Diagram	N/A

4.1.2. Flight Tracking

Table 36: Flight Tracking

Title	Flight Tracking
Description	The user will have the ability to track flight prices and find the best possible flight for their travel plans
Priority	5
Precondition(s)	Users must fill out a form detailing where they are flying to and from, and the dates in which they are planning on flying
Postconditions(s)	User is able to plan activities, book hotels and flights on the website
Use Case Diagram	N/A

4.2. Non-Functional Requirements

4.2.1. Accessibility

Table 37: Accessibility

Title	Accessibility
Description	The application should be accessible to users with disabilities, complying to accessibility standards.
Priority	4
Applicable FR(s)	N/A

4.2.2. Internalization and Localization

Table 38: Internalization and Localization

Title	Internalization and localization
-------	----------------------------------

Description	The application should support internalization and localization features to adopt content, language, currency and date formats according to the user's location
Priority	4
Applicable FR(s)	1.1.7.