Executive Summary:

The module that will be presented (which from now on shall be known as “the product”) is part of a multi-layered e-tourism online application whose goal is to simplify the process of connecting the user to travel agencies and local services based on geolocation.

The product is responsible for the gathering of data and the processing of it. The processed information will then be outputted in a easy to read and understand format that will be made available to the user.

The following subsections will make use of EARS.

Introduction:

The product is a microservice which shall collect data from travelling agencies and send this information to the rest of the modules of the application, where it shall be used to provide the user with the complete service the application is intended to offer.

The product concerns are categorized in user requirements and system requirements (both are presented in separate documents).Each of them come with different parts for completing functional and non-functional requirements.

The typical stakeholders are:

- **Producer:**
  - Requirements Team
- **Approvers:**
  - Technical Leader
  - Project Manager
  - Customer Representative(s)
- **Maintainers:**
  - Requirements Team (prior to delivery)
  - Maintenance Team (after delivery)
- **Users:**
  - Customer Organization, which uses it to understand the scope of the application to be delivered.
  - Architecture Team, which uses it to begin production of the system architectures.
  - User Experience Team, which uses it to begin production of the human interface prototype(s).
  - Independent Test Team, which uses it to begin preparation for system and launch testing.
  - Endeavor Management Team, which uses it to begin managing endeavor scope and scheduling the endeavor activities.
  - Subcontractor Organizations, organization that might use the product to drive their initial work.

Background:

The biggest design issue with monolithic application architectures is that there's so much code in them that implements widely differing functionality. To make any change to a monolithic app, you must coordinate across different groups in order to ensure that everyone's code continues operating properly. As a result, developers often spend more time on integration and testing than on delivering new application capability.

For this reason,the large application is split into different microservices who will take a part of the responsibility of the so called "main product" .The first element of a microservice is to define what it should do. In our case ,the product shall realize the communication between travel agencies and user.

Scope and Purpose:

The product shall collect the trivial information regarding points of interests , tourist attractions, leisure activities, airlines, car rentals, cruise lines, hotels, railways, travel insurance, and package tours. The regions available are Romanian Northeast + Republic of Moldova .

The user can interact with the product via most type of devices with an internet connection and can choose from the different types of available services.The users can categorize the data presented by their own personal preference.

Methods and Instrumentation:

Creativity - The generation and selection of ideas to innovate or solve a difficult problem.

Data mining - Search and filtering of requirements databases to identify relevant knowledge about stakeholder needs.

Interview meeting - Between a requirements engineer and a stakeholder to discuss topics of relevance for the system.

Introspection - Use of domain knowledge in combination with reflection and empathy to base requirements on experience.

Observation study of system use - Possibly in the target environment and by real users, to understand usage processes and strengths and weaknesses of a current system.

Limitations, Questions and Issues:

The main limitation in building the application is the conflict between simplicity/ease of use, and flexibility/powerfulness. As developers we have to find a balance between them, adding more features doesn't necessarily make the application better it can overcomplicate it and make it unusable by the common user. Also a more powerful app requires a more resources and a better internet speed to run properly.

One possible issue is that the application might not run or work as it was intended on every kind of device a user might use. Some of the problems might be incompatibility with the device(the device can be either too old or not have at least the minimum specs to run the application properly) or the application is not updated to the latest build.

Other complications related to the product and it's use are included in the Risk Assessment document.

References:

https://www.researchgate.net/publication/224079416_Easy_approach_to_requirements_syntax_EARS

https://www.w3.org/TR/2008/REC-xml-20081126/

http://www.w3.org/TR/xpath/

Appendices:

Key Terms

The following table provides definitions for terms relevant to this document.

Term	Definition
EARS	*Easy Approach to Requirements Syntax*
XML	*Extensible Markup Language*
XPath	*XML Path Language*