Software Requirements Specification

for

Outdoor Touren Portal (CrypTour)

Version 1.0

Prepared by Robin Benzinger, Fabian Kuffer, Philip Prüssner, Jannik Rasch, Kai Rösel, Victoria Strobel, David Ullmer

TINF20ITA

26.10.2021

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
CrypTour	26.10.2021	initial specification of the project	1.0

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the open-source software CrypTour. It will explain the purpose and features of the software, the interfaces of the software, what the software will do and the constraints under which it must operate. This document is intended for users of the software and potential developers.

1.2 Document Conventions

This Document was created based on the IEEE template for System Requirement Specification Documents.

1.3 Intended Audience and Reading Suggestions

- Programmers who are interested in working on the project by further developing it or fix existing bugs.
- Power-Users, who are interested in creating tours and want to know how to maximize their profit
- Regular users, who want to learn more about the software

1.4 Product Scope

CrypTour is a hub for users to create, upload, share, buy and follow hiking trails. The Product suggests different hiking trails to the user based on search criteria and allows them to buy the tour description. Buying a trail uses blockchain technology and automated market maker - tokens to complete the transaction.

1.5 References

- **IEEE Template for System Requirement Specification Documents:**
  https://goo.gl/nsUFwy

2. Overall Description

2.1 Product Perspective

CrypTour is a new and independent product and service that was conceived as a project during the software engineering course in the third semester of the IT-Automotive degree program. It is not meant to be a component of a larger system and is instead supposed to serve as the basis for a further project which will implement the automated market maker token into the product.

2.2 Product Functions

A comprehensive list of all functions is available in "Features.docx" stored in the teams team o365grpSWESTG-TINF20ITA.

2.3 User Classes and Characteristics

- Experienced Users, who already have a crypto wallet and want to use the app to find new tours and are ready to pay for it
- Power-Users, who are interested in creating and managing tours and want to maximize their profits
- Regular users, who want to discover new tours and need help following the correct trails. They may need help with using cryptocurrency and may prefer free trails.

2.4 Operating Environment

- The website works on Chromium-based Browsers.

2.5 Design and Implementation Constraints

The CrypTour website is developed in HTML, JavaScript. The database used for user management and tour information is mariadb. A REST-API is used as connection to the backend. The backend will be implemented in JavaScript.

2.6 User Documentation

The homepage of the website shows the users how using the service works by displaying tutorials for the various functionalities.

2.7 Assumptions and Dependencies

The user needs a chromium-based browser. The maps are dependent on the OpenStreetMap-API.

3. External Interface Requirements

3.1 User Interfaces

The user interface of the website is designed to be intuitive and should adjust to various aspect ratios (e.g., mobile devices).

3.2 Software Interfaces

The app can connect to a central server.

3.3 Communications Interfaces

The app requires an internet connection to find and buy the available tours. The product should allow a secure login into a user account and should synchronize between different devices with the same account.

4. System Features

This section describes CrypTour's most important and critical features and explains how a user can interact with them.

4.1 Planning a new route

Users can upload routes by uploading a gpx-file. The difficulty rating of a tour can be manually chosen by the author.

4.2 Find and acquire new tours

The User can look for new trails and apply different filters, for example length of the route and difficulty. They can then buy the tour with the market maker token system and save it to their personal collection of acquired tours.

4.3 Review tours

Every registered user can rate purchased tours and upload their own pictures for the preview view. This allows for easy identification of popular tours. Users can rate a route using a star-rating-system and write a comment about the tour. A button in the rating area allows the user to upload pictures of the tour. If an error occurs, the user should be informed with a pop-up.

4.4 Detailed description of the individual tours

Users can see detailed information about each tour. In this view, ratings from other users can also be seen. If it is a paid tour, some information can only be viewed in full after purchase, e.g., exact tour data.

4.5 Tour distribution

Tours can be provided either free of charge or for a fee. In the case of paid tours, the creator of the tour receives a share of the revenue, but the owner of the software also receives a commission. This is implemented through automated market maker token.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

CrypTour is designed to run on any modern browser.

5.2 Safety Requirements

A safety disclaimer is shown that CrypTour is not liable for any damages to the users or environment occurring during the tour.

5.3 Security Requirements

CrypTour requires access to the camera and media storage if the user wants to add a photo to a route. Users must create an account to use all features of CrypTour. Users can either login with their existing account or create a new account with an email address.

5.4 Software Quality Attributes

The website should have a simple layout so that new users can easily grasp all the functions. The Framework around the automated market maker tokens should be designed in a flexible way, so that another developer can implement the feature later.

5.5 Business Rules

Only registered users can buy and save tours.

6. Other Requirements

The software does not have to be internationalized because it will only be used in Germany.

Appendix A: Glossary

- **Automated market maker:** an underlying protocol that powers decentralized exchanges. Through autonomous trading mechanisms, centralized exchanges and related market-making techniques are eliminated.