
Software Requirments Specifikation

Bike+

Version 1.0

Prepared by:

Kerim Dincer

Cagan Yetis

24.05.2023

Table of Contents:

Revision History

Revision	Date	Authors	Changes
1.0	24.05.2023	Kerim, Cagan	Initial Version

1. Introduction

- 1.1 Purpose and Intended Audience
- 1.2 Usage of Documents
- 1.3 Acronyms and Definitions
- 1.4 SRS Structure

2. General Overview

- 2.1 Product Perspective
- 2.2 Product Capabilities
- 2.3 Data Protection
- 2.4 Operating Environment
- 2.5 Limitations

3. Interface Requirements

- 3.1 Hardware Interfaces
- 3.2 Software Interface

4. Functional Requirements

- 4.1 Calorie Deficit Calculation
 - 4.1.1 Speed And Distance
 - 4.1.2 Weight
 - 4.1.3 Slope and Incline
- 4.2 Multiple Language Choice
- 4.3 Voice Assistant
- 4.4 Multiple Road Choice
 - 4.4.1 Mountain Biking
 - 4.4.2 Road Biking
- 4.5 Connectable Devices
 - 4.5.1 BT Chestbelt
 - 4.5.2 RPM of Pedals

5. Nonfunctional Requirements

- 5.1 Interface Quality
 - 5.1.1 Simplicity
 - 5.1.2 Understandability
- 5.2 Performance Requirements
 - 5.2.1 Fast Processing time
 - 5.2.2: Error-freeability
- 5.3 Sustainability
 - 5.3.1 Energy Efficiency

6. Use Cases

- 6.1 Use Case Diagramm
- 6.2 Use Case Description

1.Introduction

1.1 Purpose and Intended Audience

This software will be designed for the advanced bikers. Although even a beginner can use this software. This app can only be used on smart watches. It cannot be used on other devices at this time.

1.2 Usage of Documents

The user of 'bike+' will be referred to as the 'Biker'. Functional requirements will be rated as low, medium, and high. Functional requirements may be interrelated.

1.3 Acroynms and Definitons

In this section, some of the terms we use will be explained and abbreviated.

- SRS - System Requirements Specifications
- UCD - Use Case Diagramm,

UCD Defination: A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well.

- RPM - Round Per Minute
- GPS - Global Positioning System
- BT - Bluetooth

1.4 SRS Structure

SRS, lists functional and non-functional requirements of 'bike+'. And explain them to 'biker' in a meaningful way.

It also gives comprehensive information about the functionality of the application.

2. General Overview

2.1 Product Perspective

The 'bike+' app has a wide target audience. People of all ages can easily adapt to this application. There is no specific age group targeted by the application. The vision of the application is to provide an optimal user experience to the 'biker'. The user interface is customized in this respect. The aim is to provide the 'biker' with a long-term 'bike assistant'.

2.2 Product Capabilities

The application provides the biker with a lot of data about his/her training or himself/herself. These are parameters such as calorie calculation, heart rate, rpm, distance traveled, speedometer.

The biker can switch between 2 modes within the app according to his/her biking style. These are mountain biking and road biking. About road biking, the biker can follow the data about the roads from the application interface by means of gps.

In addition, the application has German and English language options.

2.3 Data Protection

The application will not connect to any other database. The most important reason for this is the importance given to data protection. According to the feedback we received from the user, the connections of the application will only be the bluetooth chest belt and rpm sensors on the pedals.

2.4 Operating Environment

The application is a smart watch-specific application. It is compatible with Android and IOS-based wearable technology devices.

It requests the location information of the biker in order to work optimally. However, the location records are not stored in any database, so the confidentiality of the data is at the forefront for the application.

2.5 Limitations

Our product will be used as a wearable application for a smart watch. And for this it must meet some requirements. These requirements are internet connection, downloadability. For the application to work optimally, the devices running the application must require these specifications.

3.Interface Requirements

3.1 Hardware Interfaces

A hardware interface is required for the smartwatch to communicate with the chest belt. Profiles such as BLE (Bluetooth low energy) and HRP (Heart Rate Profile) are used for receiving and sending heart rate data. The hardware interface allows the smartwatch app to communicate with the chest belt correctly.

3.2 Software Interfaces

The application needs a user interface to interact with the user. This is used to show information to the user, respond to buttons or touchscreen controls, and navigate within the app.

4. Functional Requirements

4.1 Calorie Deficit Calculation

The application should be able to calculate the amount of calories consumed by the biker while training, through parameters such as speed and distance, weight, slope and incline.

4.1.1 Speed And Distance

Requirements	Speed and Distance
Explanation	The app needs the distance and speed taken to calculate the calorie deficit.
Priority	High

4.1.2 Weight

Requirements	Weight
Explanation	The biker's weight affects the number of calories burned while cycling. A heavier person burns more calories than a lighter person doing the same activity. That is why the app needs weight data.
Priority	High

4.1.3 Slope and Incline

Requirements	Slope and Incline
Explanation	Cycling on a sloping road affects the biker's calorie burn. As the degree of incline and incline increases, the amount of calories expended increases.
Priority	High

4.2 Multiple Language Choice

In order for the biker to use the application more comfortably, two different language options are offered.

Requirements	Multiple Language Choice
Explanation	There are only 2 language options. These are German and English
Priority	Medium

4.3 Voice Assistant

The voice assistant provides the biker with a better experience while driving.

Requirements	Voice Assistant
Explanation	During the riding, biker can ask informations about the ride.(Map,Speed,Hearbeat,etc)
Priority	Low

4.4 Multiple Road Choice

There are 2 different modes that can be selected according to the driving style within the application. These are mountain biking and road biking.

4.4.1 Mountain Biking

If mountain mode is selected during use, a suitable interface and precautions to be taken by the biker are provided to the user.

Requirements	Mountain Biking
Explanation	Possible hazards arising from the road (such as bumps, pits) are provided to the biker.
Priority	High

4.4.2 Road Biking

If road biking mode is selected during use, a suitable interface and precautions to be taken by the biker are provided to the user.

Requirements	Road Biking
Explanation	The user is provided with the safest and fastest possible route.

Priority	High
----------	------

4.5 Connectable Devices”

There are 2 different compitable devices with application. These are BT chess belt and RPM cencor on pedals.

4.5.1 BT Chest Belt

Chest belt is fully compatible with the application. It gives the user instant heart rate data.

Requirements	BT Chest Belt
Explanation	Biker can see the number of heart beats per minute via the BT chest belt.
Priority	High

4.5.2 RPM of Pedals

The biker can see the speed and distance information and the RPM number.

Requirements	RPM of Pedals
Explanation	Biker can see RPM via the BT chest belt.
Priority	High

5. Nonfunctional Requirements

5.1 Interface Quality

The application interface will be simple and understandable according to the user's wishes.

5.1.1 Simplicity

In this application there will be no redundant attribute in interface that confuses user. According to the user's wishes interface will be super simple.

Requirements	Simplicity
Explanation	There will be no redundant informations on the surface.
Priority	High

5.1.2 Understandability

In this application there will be no software jargon and that will help user to understand the application easily.

Requirements	Understandability
Explanation	Software jargon will be avoided as much as possible in the application so that the user does not experience difficulties while using the application.
Priority	High

5.2 Performance Requirements

Application performance will be kept at the highest level so that the application can run as fast and error-free as possible.

5.2.1 Fast Processing time

Requirements	Fast Processing time
Explanation	The application should run as fast as possible. Because we want the user to get the fastest response in possible situations that he/she may encounter while using the application.
Priority	medium

5.2.2: Error-freebility

Since the user usually uses the application during training, this topic is important as any error encountered in the application may affect the driving experience.

Requirements	Error-freebility
Explanation	The application should run as error-free as possible. It is not desired for users to encounter a negative situation.
Priority	high

5.3 Sustainability

In order for the application to be used for many years, easy maintenance and updates are required. These are essential for sustainability.

5.3.1 Energy Efficiency

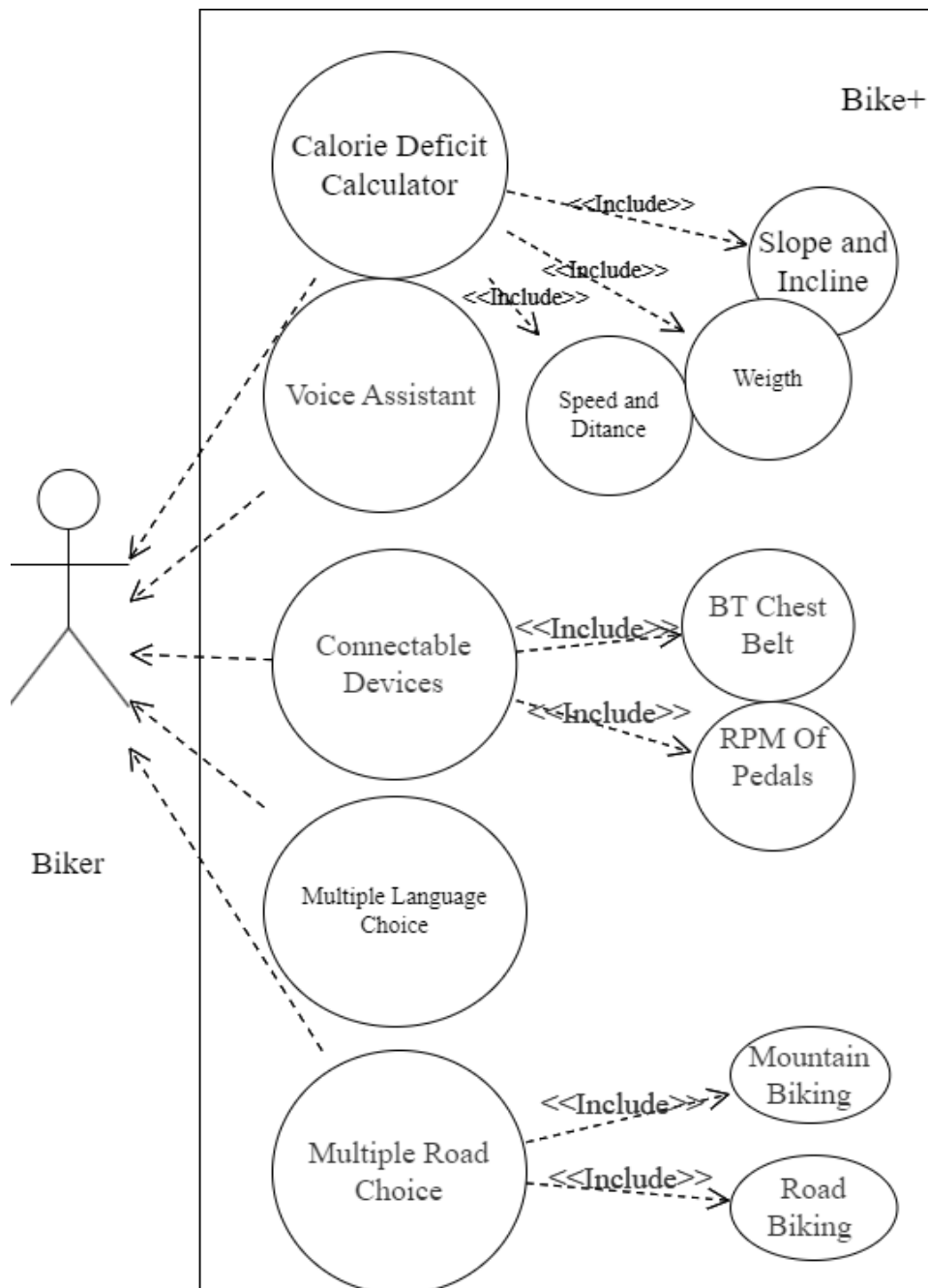
The energy-efficient operation of the application ensures that the battery life is extended and energy consumption is reduced.

Requirements	Energy Efficiency
Explanation	Optimizing the processes running in the background prevents the device from wasting energy unnecessarily.
Priority	high

6. Use Cases

This Chapter provides a summary of all use cases of Bike+.

6.1 Use Case Diagramm



6.2 Use Case Description

Detailed description of the individual Use Cases:

Name	Calorie Deficit Calculation
ID	UC1
Aim	The goal is to calculate the calorie deficit of the biker.
Precondition	Application needs to know Speed and Distance, Slope and Incline, Weight to calculate calorie deficit consistent.
Postcondition	Calculates the calorie deficit.
Postcondition in case of exception	Calorie deficit is calculated incorrectly.
Actors	Biker.
Normal Process	<ol style="list-style-type: none">1. User enters his/her weight.2. App calculates speed and distance traveled.3.App calculates slope and incline.4. Using this information, the calorie deficit is calculated.
Exceptions	<ol style="list-style-type: none">1a If the user enters a value less than 0 for the weight, it will give an error.

Name	Multiple Language Choice
ID	UC2
Aim	The application works in different languages.
Precondition	-
Postcondition	The application works in 2 different languages, English and German.
Postcondition in case of exception	-
Actors	Biker.
Normal Process	<ol style="list-style-type: none"> 1. The biker opens the application. 2. The biker selects one of the two language options.
Exceptions	-

Name	Voice Assistant
ID	UC3
Aim	Provides a voice assistant to assist the cyclist while riding
Precondition	User must allow a voice assistant
Postcondition	Voice assistant will work.
Postcondition in case of exception	The voice assistant will not work if the Input and Output devices are not allowed to work.
Actors	Biker.
Normal Process	<ol style="list-style-type: none"> 1. Biker allows the Input and Output devices. 2. Biker gives command to voice assistant to work. 3. The biker asks the voice assistant about the subject she/he wants to be helped.
Exceptions	<p>1a The voice assistant will not work if the user does not accept the required permissions</p> <p>2a Voice assistant will not work if English and non-german language are used</p>

Name	Multiple Road Choice
ID	UC4
Aim	The user chooses one of two different road options.
Precondition	-
Postcondition	One of two different interfaces opens in the application.
Postcondition in case of exception	-
Actors	Biker.
Normal Process	-
Exceptions	-

Name	Mountain Biking
ID	UC4.1
Aim	It is intended to provide an interface to assist the biker during mountain riding.
Precondition	The user needs to select the mountain biking mode.
Postcondition	The app adapts to mountain biking.
Postcondition in case of exception	-
Actors	Biker.
Normal Process	<ol style="list-style-type: none"> 1. User selects mountain biking interface. 2. It uses an interface suitable for mountain driving.
Exceptions	1a If the user selects the road biking mode, they cannot access the mountain biking mode.

Name	Road Biking
ID	UC4.2
Aim	It is intended to provide an interface to assist the biker during road riding.
Precondition	The user needs to select them road biking mode.
Postcondition	The app adapts to road biking.
Postcondition in case of exception	-
Actors	Biker.
Normal Process	<ol style="list-style-type: none"> 1. User selects road biking interface. 2. It uses an interface suitable for road driving.
Exceptions	1a If the user selects the mountain biking mode, they cannot access the road biking mode.

Name	Connectable Devices
ID	UC5
Aim	It is aimed that the user can see the desired data while biking.
Precondition	BT Connection.
Postcondition	User can see his/her data such as heartbeat.
Postcondition in case of exception	The user cannot see his/her data during ride.
Actors	Biker.
Normal Process	-
Exceptions	-

Name	BT Chest Belt
ID	UC5.1
Aim	It is aimed that the user can see his/her heartbeat per minute while biking.
Precondition	BT Connection, BT Chestbelt
Postcondition	User can see his/her data such as heartbeat.
Postcondition in case of exception	The user cannot see his/her data such as heartbeat.
Actors	Biker.
Normal Process	<ol style="list-style-type: none"> 1. BT Chestbelt connects to the application. 2. Biker starts to riding. 3. BT Chestbelt gets data from the biker. 4. Heartbeat per minute is being calculated by the application.
Exceptions	1a In condition of BT Chesbelts not getting paired with device that biker uses, heartbeat will not be calculated.

Name	BT RPM of Pedals
ID	UC5.2
Aim	It is aimed that the user can see his/her RPM while biking.
Precondition	BT Connection
Postcondition	User can see his/her RPM while riding.
Postcondition in case of exception	The user cannot see his/her RPM while riding.
Actors	Biker.
Normal Process	<ol style="list-style-type: none"> 1. Biker starts to riding. 2. Cencors in pedals gets data from the biker. 3. RPM will be calculated.
Exceptions	1a in some cases the sensors may not work or give incorrect information. In this case, the data given to the user by the application will be wrong with this ratio.