SRS Document
SnakeTracker

1. Introduction

1.1 Purpose of the document
The purpose of this document is to build a system that lets you create and manage snakes to have an easy method for breeders and zoos to keep track of their snakes.

1.2 Intended audience
This project is aimed at snake breeder, zoos and private keeper of snakes.

1.3 Project scope
The purpose of the SnakeTracker is to have a clear view of your existing snakes and their feeding cycles. It lets you easily update data like weight or the last time you fed your snake. It is supported by a MySQL Database.

2. Overall description

2.1 Product perspective
The SnakeTracker database stores the following information

- Snake details:
  It includes the type, name, sex, birthdate and an image path to a locally saved image of your snake.

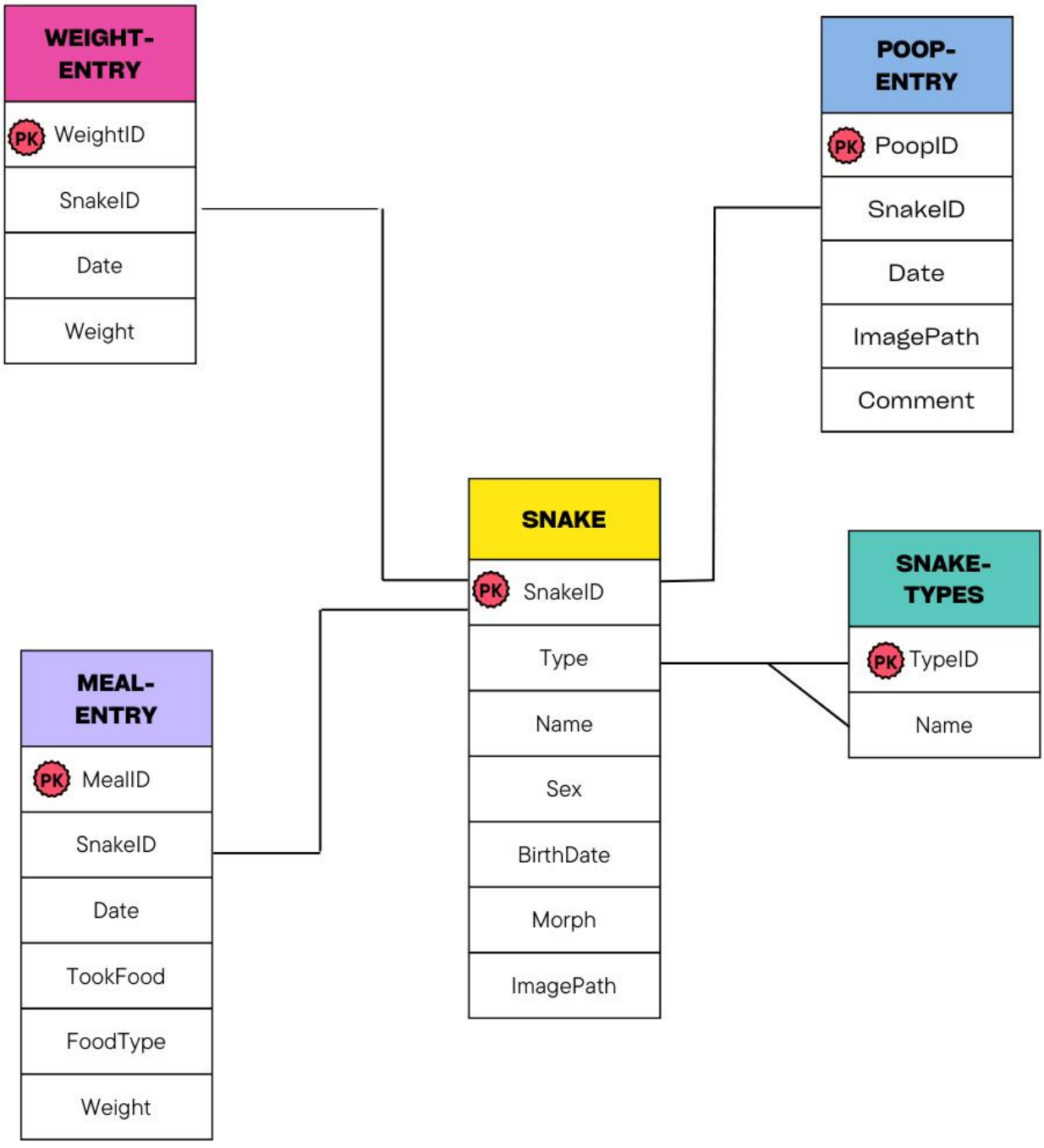
- Weight details:
  It includes the weight in kg and the date of the data input

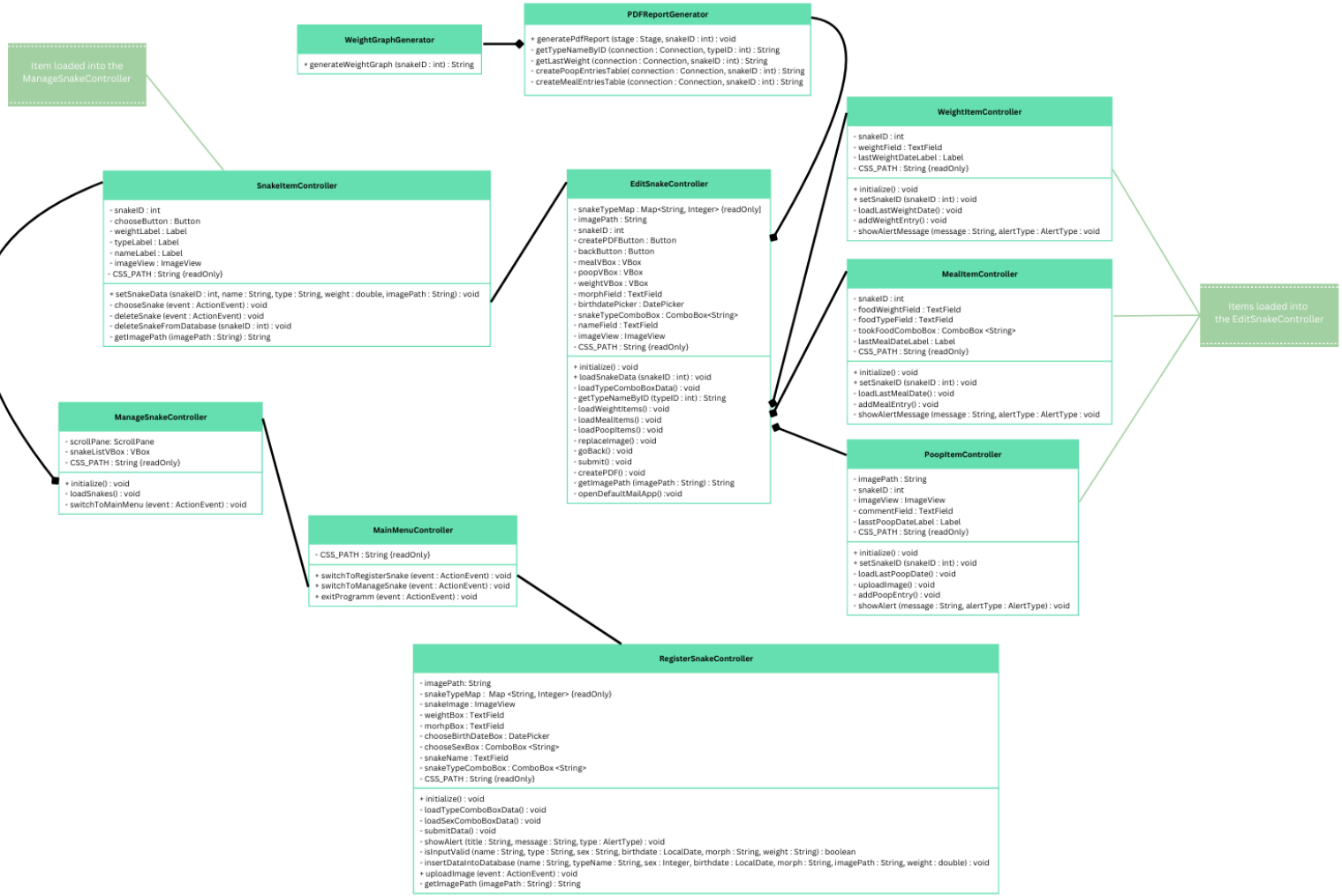
- Meal details:
  It includes if the offered food was taken, weight and what kind of meal and the date of the input.

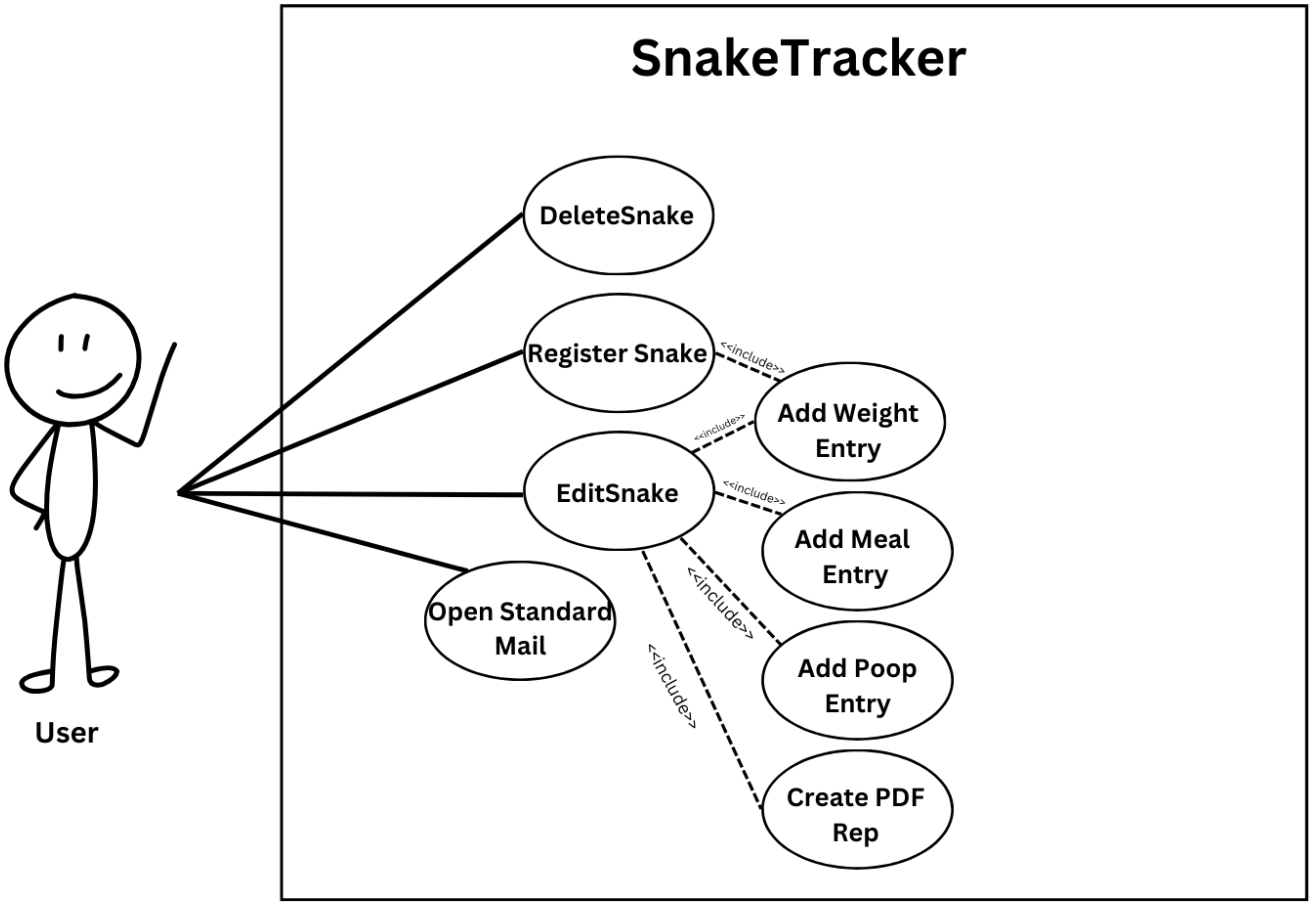
- Poop details:
  It includes a comment for example if there are any health issues regarding the poop/snake, an image path if you decide to upload an image of the poop and the date of the input.

2.2 Product Features

The major features of the SnakeTracker database as shown in the ER model, classdiagram and Use-Case-Diagramm below.




Item loaded into the ManageSnakeController
WeightGraphGenerator
+ generateWeightGraph (snakeID : int) : String
PDFReportGenerator
+ generatePdfReport (stage : Stage, snakeID : int) : void
- getTypeNameByID (connection : Connection, typeID : int) : String
- getLastWeight (connection : Connection, snakeID : int) : String
- createPoopEntriesTable(connection : Connection, snakeID : int) : String
- createMealEntriesTable (connection : Connection, snakeID : int) : String
WeightItemController
- snakeID : int
- weightField : TextField
- lastWeightDateLabel : Label
- CSS_PATH : String (readOnly)
+ initialize() : void
+ setSnakeID (snakeID : int) : void
- loadLastWeightDate() : void
- addWeightEntry() : void
- showAlertMessage (message : String, alertType : AlertType : void
SnakeItemController
- snakeID : int
- chooseButton : Button
- weightLabel : Label
- typeLabel : Label
- nameLabel : Label
- imageView : ImageView
- CSS_PATH : String (readOnly)
+ setSnakeData (snakeID : int, name : String, type : String, weight : double, imagePath : String) : void
- chooseSnake (event : ActionEvent) : void
- deleteSnake (event : ActionEvent) : void
- deleteSnakeFromDatabase (snakeID : int) : void
- getImagePath (imagePath : String) : String
EditSnakeController
- snakeTypeMap : Map<String, Integer> (readOnly)
- imagePath : String
- snakeID : int
- createPDFButton : Button
- backButton : Button
- mealVBox : VBox
- poopVBox : VBox
- weightVBox : VBox
- morphField : TextField
- birthdatePicker : DatePicker
- snakeTypeComboBox : ComboBox<String>
- nameField : TextField
- imageView : ImageView
- CSS_PATH : String (readOnly)
+ initialize() : void
+ loadSnakeData (snakeID : int) : void
- loadTypeComboBoxData() : void
- getTypeNameByID (typeID : int) : String
- loadWeightItems() : void
- loadMealItems() : void
- loadPoopItems() : void
- replaceImage() : void
- goBack() : void
- submit() : void
- createPDF() : void
- getImagePath (imagePath : String) : String
- openDefaultMailApp() :void
MealItemController
- snakeID : int
- foodWeightField : TextField
- foodTypeField : TextField
- tookFoodComboBox : ComboBox <String>
- lastMealDateLabel : Label
- CSS_PATH : String (readOnly)
+ initialize() : void
+ setSnakeID (snakeID : int) : void
- loadLastMealDate() : void
- addMealEntry() : void
- showAlertMessage (message : String, alertType : AlertType : void
Items loaded into the EditSnakeController
ManageSnakeController
- scrollPane: ScrollPane
- snakeListVBox : VBox
- CSS_PATH : String (readOnly)
+ initialize() : void
- loadSnakes() : void
- switchToMainMenu (event : ActionEvent) : void
PoopItemController
- imagePath : String
- snakeID : int
- imageView : ImageView
- commentField : TextField
- lasstPoopDateLabel : Label
- CSS_PATH : String (readOnly)
+ initialize() : void
+ setSnakeID (snakeID : int) : void
- loadLastPoopDate() : void
- uploadImage() : void
- addPoopEntry() : void
- showAlert (message : String, alertType : AlertType) : void
MainMenuController
- CSS_PATH : String (readOnly)
+ switchToRegisterSnake (event : ActionEvent) : void
+ switchToManageSnake (event : ActionEvent) : void
+ exitProgramm (event : ActionEvent) : void
RegisterSnakeController
- imagePath: String
- snakeTypeMap : Map <String, Integer> (readOnly)
- snakeImage : ImageView
- weightBox : TextField
- morhpBox : TextField
- chooseBirthDateBox : DatePicker
- chooseSexBox : ComboBox <String>
- snakeName : TextField
- snakeTypeComboBox : ComboBox <String>
- CSS_PATH : String (readOnly)
+ initialize() : void
- loadTypeComboBoxData() : void
- loadSexComboBoxData() : void
- submitData() : void
- showAlert (title : String, message : String, type : AlertType) : void
- isInputValid (name : String, type : String, sex : String, birthdate : LocalDate, morph : String, weight : String) : boolean
- insertDataIntoDatabase (name : String, typeName : String, sex : Integer, birthdate : LocalDate, morph : String, imagePath : String, weight : double) : void
+ uploadImage (event : ActionEvent) : void
- getImagePath (imagePath : String) : String



3. System Features

3.1 Functional requirements

Requirement ID	1
Requirement Name	“Register Snake”
Requirement Description	As a user you can register a snake. The registered snake will be submitted into database and if the user chose to upload an image this will be saved locally.
Type of Requirement	functional
Status	Implemented

Requirement ID	2
Requirement Name	“Edit Snake”
Requirement Description	As a user you can edit your snakes. While editing you are also able to create different entries like weight, meal and poop. Furthermore, you can create a PDF report, which includes a weight graph of last weight entries.
Type of Requirement	functional
Status	Implemented

Requirement ID	3
Requirement Name	“Add Weight Enry”
Requirement Description	As a user you create your first weight entry when register a snake. Afterwards you are able to create more weight entries while editing your snake. With the data collected the PDF report will also have a graph showing the growth of your snake's weight.
Type of Requirement	functional
Status	Implemented

Requirement ID	4
Requirement Name	“Add Meal Entry”
Requirement Description	As a user you can create meal entries while editing your snake. You can submit data like if the snake took the food or not, what kind of food it got and how heavy the meal was.
Type of Requirement	functional
Status	Implemented

Requirement ID	5
Requirement Name	“Add Poop Entry”
Requirement Description	As a user you can create poop entries while editing your snake. With every entry you can submit a picture of said poop and write a comment for example regarding health
Type of Requirement	functional
Status	Implemented

Requirement ID	6
Requirement Name	“Create PDF Report”
Requirement Description	As a user you can create a PDF report. This report includes the latest data of your snake and a graph showing the weight gain of your snake
Type of Requirement	functional
Status	Implemented

Requirement ID	7
Requirement Name	“Register Snake”
Requirement Description	As a user you can delete snakes entirely from the database. For example, if you sold the snake or if it passed away.
Type of Requirement	functional
Status	Implemented

Requirement ID	8
Requirement Name	“Open Standard Mail”
Requirement Description	As a user you can open your standard mail app so you are able to quickly send your PDF reports to others.
Type of Requirement	functional
Status	Implemented