SOFTWARE REQUIREMENTS SPECIFICATION

for

Airbnb Data Mart

Contents

1 Introduction

1.1 Purpose

The main purpose of this project and the goal of the software solution described in this document is to deepen and solidify knowledge regarding the use of the SQL programming language and relational databases in general.

1.2 Scope

In the scope of this project a 'Data Mart' for temporary renting of private owned lodgings, much like the well-known web platform Airbnb should be designed and implemented. To test the system, the database should be populated with appropriate self developed sample data. Additionally the software solution should be developed and documented according to the guidelines of the portfolio project.

1.3 Product Overview

1.3.1 Product Perspective

In the context of the described project, the data mart is a single MySQL database which is integrated in a development environment featuring a local server and an administration tool for developing and testing. The data mart is thereby self-contained and can be imported in every environment with the appropriate set-up. For the integration described in this document, the XAMPP software package is used which is a Apache distribution containing MariaDB and PHP. The software suite is available for Windows, Linux and OSX. Additionally XAMPP features phpMyAdmin as an administration tool for MySQL databases. phpMyAdmin is therefore the main user interface for the data mart. The data mart is controlled through SQL commands which are issued through the user interface. For most functionality pre-written SQL commands are stored in views and routines inside the user interface. This allows for convenient usage and testing of the data mart.

1.3.2 Product Functions

The functionality of the data mart can be derived from the temporal renting use case. First of all a feature for user registration which stores personal information and distinguishes between guest and host users is implemented. Following is the option for a host user to issue a detailed description of a lodging and post it. A guest user can use the posted lodgings to make a booking over a certain timespan and write a review about the lodging. For financial purposes a feature for processing information about transactions which supports multiple currency conversions is integrated. In addition to the core features for rental, functionality for searching nearby public transport and sights is added.

2 Requirements

To ensure a working and efficient data mart, a diverse set of functional requirements must be satisfied. This results in a database schema featuring all tables with correct data types, constraints and relations to enable the functionality specified in the product functions. Cardinalities are enforced through the constraints specified during table creation. In addition the database has an adequate level of normalization to minimize storage of redundant data.

2.1 User Management

Like already mentioned, are two types of user possible with a user taking either the role as 'Guest' or 'Host'. Both types of user are stored in the users table. For a guest user the table must contain the unique user-name, personal information like first and last name, contact information like email and phone number and the user-specific standard currency. The host user has the same attributes as the guest user with additional attributes like an optional about text and a flag that the user is in fact a host. The host user identification is realised with a recursive relation enforced through a foreign key constraint. Thereby the foreign key references the primary key of the user table which allows for filtering host users with a join statement.

2.2 Posting and Searching Lodgings

To guarantee a good user experience, lodgings must be described detailed and accurate. To accurately describe the lodging various attributes can and must be specified. This includes textual descriptions and the specification of physical attributes like rooms. Additionally price information and the rating calculated from all guest reviews is stored. The lodging table is related to the central location table where all location data is stored. Through the many detailed options a guest can easily find an appropriate lodging by leveraging the many attributes as search criteria.

2.3 Booking and Transaction System

For booking a stay in a lodging, a user can make a query where an arrival and departure date is issued. Then the data mart checks if the lodging is not already booked and depending on the case, inserts an entry in the booking table, which is connected to the lodging table. Every booking also has exactly one related entry in the transaction table, which represents the financial information of the booking. When a guest books a stay, the most recent exchange rate of the lodging's currency and the guest's currency is stored. This allows to check if the guest paid the right amount of money even when the live exchange rate which is stored in the currency table has changed. Additionally a transaction has boolean attributes, for displaying the transaction status. Thereby the attribute 'Received' is set true when the guest pays the amount due and the attribute 'Settled' is set true if the money is paid out to the host. The host can look at his earnings by querying all unsettled transaction. The earnings are always the amount subtracted by the five percent commission fee which the guest pays for using the service.

2.4 Location System

Every lodging in the data mart is posted with the exact geographical data which includes longitude and latitude. Given that the data mart also features geographical data of selected sights and public transport, the user can search for nearby locations and get a calculated distance from his lodging.

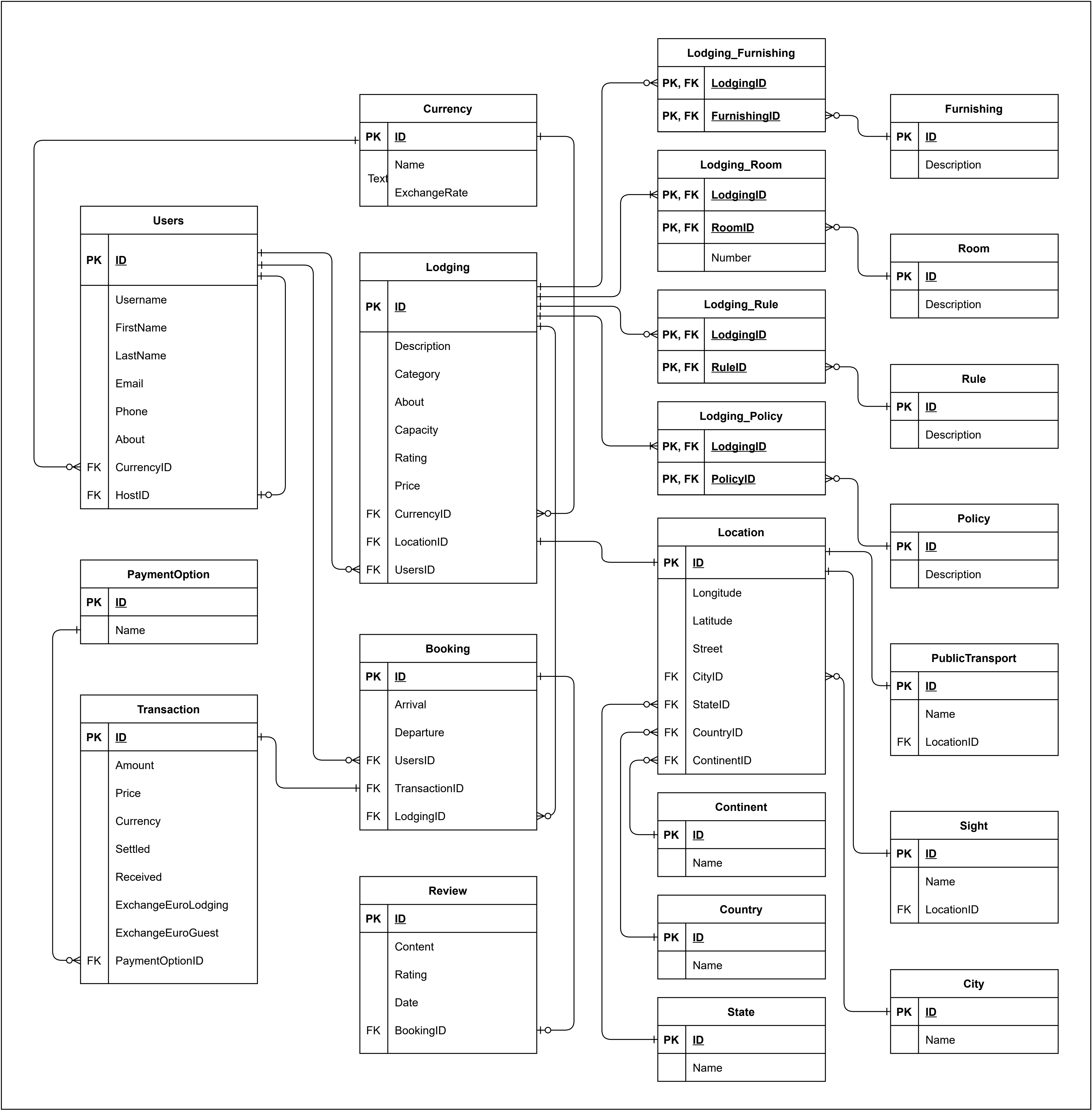
Currency
PK ID
Name
Text
ExchangeRate
Users
PK ID
Username
FirstName
LastName
Email
Phone
About
FK CurrencyID
FK HostID
Lodging
PK ID
Description
Category
About
Capacity
Rating
Price
FK CurrencyID
FK LocationID
FK UsersID
Lodging_Furnishing
PK, FK LodgingID
PK, FK FurnishingID
Furnishing
PK ID
Description
Lodging_Room
PK, FK LodgingID
PK, FK RoomID
Number
Room
PK ID
Description
Lodging_Rule
PK, FK LodgingID
PK, FK RuleID
Rule
PK ID
Description
Lodging_Policy
PK, FK LodgingID
PK, FK PolicyID
Policy
PK ID
Description
Location
PK ID
Longitude
Latitude
Street
FK CityID
FK StateID
FK CountryID
FK ContinentID
PublicTransport
PK ID
Name
FK LocationID
PaymentOption
PK ID
Name
Booking
PK ID
Arrival
Departure
FK UsersID
FK TransactionID
FK LodgingID
Transaction
PK ID
Amount
Price
Currency
Settled
Received
ExchangeEuroLodging
ExchangeEuroGuest
FK PaymentOptionID
Review
PK ID
Content
Rating
Date
FK BookingID
Continent
PK ID
Name
Country
PK ID
Name
State
PK ID
Name
Sight
PK ID
Name
FK LocationID
City
PK ID
Name

Table Name	Attribute Name	Content	Type	Constraints	PK/FK
Users	ID	PK of the Users table	INT	NOT NULL AUTO_INCREMENT	PK
	Username	Unique username of a customer	VARCHAR(128)	NOT NULL UNIQUE	
	FirstName	First name of a customer	VARCHAR(128)	NOT NULL	
	LastName	Last name of a customer	VARCHAR(128)	NOT NULL	
	Email	Customer email	VARCHAR(128)	NOT NULL	
	Phone	Customer phone number	VARCHAR(20)	NOT NULL	
	About	Short description for a host	TEXT		
	CurrencyID	Currency for price display and payments	INT	NOT NULL	FK
	HostID	Recursive FK used to identify a host user	INT		FK
Lodging	ID	PK of the Lodging table	INT	NOT NULL AUTO_INCREMENT	PK
	Description	Short description of the lodging with key details	VARCHAR(128)	NOT NULL	
	Category	Lodging category, for example appartment, flat, etc	VARCHAR(64)	NOT NULL	
	About	Detailed description about the lodging	TEXT	NOT NULL	
	Capacity	Maximal number of residents	INT	NOT NULL	
	Rating	Average rating calculated from all guest reviews	DECIMAL(2,1)		
	Price	Price per night in the hosts chosen currency	DECIMAL(7,2)	NOT NULL	
	CurrencyID	Currency of the price per night	INT	NOT NULL	FK
	LocationID	Junction to the Location table, location of the Lodging	INT	NOT NULL	FK
	UsersID	Junction to the Users table, Host/Owner of the Lodging	INT	NOT NULL	FK
Currency	ID	PK of the Currency table	INT	NOT NULL AUTO_INCREMENT	PK
	Name	Name of the currency	VARCHAR(64)	NOT NULL UNIQUE	
	ExchangeRate	Exchange rate of the currency to Euro (Updated regularly)	DECIMAL(20,9)	NOT NULL	
Booking	ID	PK of the Booking table	INT	NOT NULL AUTO_INCREMENT	PK
	Arrival	Date of the check-in / arrival	DATE	NOT NULL	
	Departure	Date of the check-out / departure	DATE	NOT NULL	
	UsersID	Junction to the Users table	INT	NOT NULL	FK
	TransactionID	Junction to the Transaction table	INT	NOT NULL	FK
	LodgingID	Junction to the Lodging table	INT	NOT NULL	FK
Transaction	ID	PK of the Transaction table	INT	NOT NULL AUTO_INCREMENT	PK
	Amount	Summed price of the booking plus 5% provision	DECIMAL(19,2)	NOT NULL	
	Price	Lodging price per night at the time of booking	DECIMAL(19,2)	NOT NULL	
	Currency	Chosen currency of the guest	VARCHAR(64)	NOT NULL	
	Received	Flag if the money has arrived on the bank account	BOOLEAN	NOT NULL DEFAUT FALSE	
	Settled	Flag if the money was paid to the host	BOOLEAN	NOT NULL DEFAUT FALSE	
	ExchangeEuroLodging	Exchange rate from lodging currency to Euro on the day of payment	DECIMAL(20,9)	NOT NULL	
	ExchangeEuroGuest	Exchange rate from Euro to guest currency on the day of payment	DECIMAL(20,9)	NOT NULL	
	PaymentOptionID	User chosen payment option for the transaction	INT	NOT NULL	FK

Table Name	Attribute Name	Content	Type	Constraints	PK/FK
Review	ID	PK of the Review table	INT	NOT NULL AUTO_INCREMENT	PK
	Content	Content of the review written by a guest	TEXT	NOT NULL	
	Rating	Rating of the lodging from 1-5 stars	DECIMAL(2,1)	NOT NULL	
	BookingID	Junction to the booking which corresponds to the review	INT	NOT NULL	FK
Furnishing	ID	PK of the Furnishing table	INT	NOT NULL AUTO_INCREMENT	PK
	Description	Type of furnishing e.g. fridge, air conditioning, wlan, etc	VARCHAR(64)	NOT NULL	
Lodging_Furnishing	LodgingID	Junction to the Lodging table	INT	NOT NULL	PK/FK
	FurnishingID	Junction to the Furnishing table	INT	NOT NULL	PK/FK
Room	ID	PK of the Room table	INT	NOT NULL AUTO_INCREMENT	PK
	Description	Description of the room e.g. bathroom, livingroom, etc	VARCHAR(64)	NOT NULL	
Lodging_Room	LodgingID	Junction to the Lodging table	INT	NOT NULL	PK/FK
	RoomID	Junction to the Room table	INT	NOT NULL	PK/FK
	Number	Number of a specific room in a lodging, e.g 2 bathrooms	TINYINT	NOT NULL	
Rule	ID	ID of the Rule table	INT	NOT NULL AUTO_INCREMENT	PK
	Description	Description of a house rule, e.g. No smoking, no pets, check-in time	VARCHAR(64)	NOT NULL	
Lodging_Rule	LodgingID	Junction to the Lodging table	INT	NOT NULL	PK/FK
	RuleID	Junction to the Rule table	INT	NOT NULL	PK/FK
Policy	ID	PK of the Policy table	INT	NOT NULL AUTO_INCREMENT	PK
	Description	Contains information like damage clauses, cancellation conditions, etc	TEXT	NOT NULL	
Lodging_Policy	LodgingID	Junction to the Lodging table	INT	NOT NULL	PK/FK
	PolicyID	Junction to the Policy table	INT	NOT NULL	PK/FK
Location	ID	PK of the Location table	INT	NOT NULL AUTO_INCREMENT	PK
	Longitude	Longitude of a location	DECIMAL(7,5)	NOT NULL	
	Latitude	Latitude of a location	DECIMAL(7,5)	NOT NULL	
	Street	Street name and house number/flat number, etc.	VARCHAR(128)	NOT NULL	
	CityID	City or Town of the location	INT	NOT NULL	FK
	StateID	State or Province of the location	INT	NOT NULL	FK
	CountryID	Country of the location	INT	NOT NULL	FK
	ContinentID	Continent of the location	INT	NOT NULL	FK
Continent	ID	PK of the Continent table	INT	NOT NULL AUTO_INCREMENT	PK
	Name	Name of the continent	VARCHAR(16)	NOT NULL	

Table Name	Attribute Name	Content	Type	Constraints	PK/FK
Country	ID	PK of the Country table	INT	NOT NULL AUTO_INCREMENT	PK
	Name	Name of the country	VARCHAR(64)	NOT NULL	
State	ID	PK of the State table	INT	NOT NULL AUTO_INCREMENT	PK
	Name	Name of the state or province	VARCHAR(128)	NOT NULL	
City	ID	PK of the City table	INT	NOT NULL AUTO_INCREMENT	PK
	Name	Name of the city/town	VARCHAR(128)	NOT NULL	
Sight	ID	PK of the Sight table	INT	NOT NULL AUTO_INCREMENT	PK
	Name	Name of the sight	VARCHAR(128)	NOT NULL	
	LocationID	Location of the sight	INT	NOT NULL	FK
PublicTransport	ID	PK of the PublicTransport table	INT	NOT NULL AUTO_INCREMENT	PK
	Description	Description of the transport, e.g. central train station, bus stop, etc	VARCHAR(64)	NOT NULL	
	LocationID	Location of the public transport	INT	NOT NULL	FK
PaymentOption	ID	PK of the PaymentOption table	INT	NOT NULL AUTO_INCREMENT	PK
	Name	Name of the payment option, e.g. Paypal, Credit Card	VARCHAR(64)	NOT NULL UNIQUE	