Software Requirements Specification

for

Amusement Park Management

Team Hydrohomies
Jaiwanth Mandava
Kalyan Adithya
Vivek Pamnani

October 14, 2019

Chapter 1

Introduction

1.1 Purpose

The system is to maintain a record of the events that happen in an Amusement Park, like tourist enjoying rides and meals, staff recruiting and firing.

1.2 Intended Audience and Reading Suggestions

The system is intended for observers (to get reports), staff to view information and managers to view/edit information.

Chapter 2

Database Requirements

2.1 Strong Entity Types

Employee

- Name
  - First Name
  - Last Name
- SSN (Key Attribute)
- Birthday
- Age (Derived Attribute)
- Address
  - Street
  - Area
  - City

Visitor

- Name
  - First Name
  - Last Name
- Number (Key Attribute)
- Address
  - Street
  - Area
  - City
- Age

Rides

- ID (Key Attribute)
- Name
- Cost
- VIP

Mess

- Name (Key Attribute)
- Vegetarian
- Price
- Items (Multivalued Attribute)

Shops

- Name
- Shop License (Key Attribute)
- Open
- Type

2.2 Weak Entity Types

Card

- Card ID (Discriminator/Partial Key)
- Balance
- VIP

Identified by Visitor entity type.

Feedback

- Feedback Number (Discriminator/Partial Key)
- Rating
- Review

Identified by Visitor entity type.

2.3 Relationship Types

Enjoys

Maps Visitor to Rides
A many-many relationship with complete participation of Visitor entity type.

Eats

Maps Visitors to Mess
A many-one relationship with complete participation of Visitor entity type.

Buys

Maps Visitors to Shops
A many-many relationship.

Owns

Maps Employees to Shops.
A one-one relationship with complete participation of Shops entity type.

Works_At

Maps Employees to Mess
A one-one relationship with complete participation of Mess entity type.

Works_On

Maps Employees to Rides
A many-one relationship with complete participation of Rides entity type.

Holds

Maps Visitor to Card
A one-many identifying relationship with complete participation of Visitor entity type.

Gives

Maps Visitor to Feedback
A one-many identifying relationship with complete participation of Visitor entity type.

2.4 An n = 4 n-ary relationship

Uses

Relates the following entities:

- Visitor

- Card
- Feedback
- Ride (Cardinality: 1, i.e, <Visitor, Card, Feedback> is mapped to a single entity Ride)

2.5 Subclasses

Manager

Subclass of the superclass Employee.

Technician

Subclass of the superclass Employee.

Janitor

Subclass of the superclass Employee.

Attendee

Subclass of the superclass Employee.

Chapter 3

Functional Requirements

3.1 Observer Interfaces

These functions are available to any observer (included in all following interfaces).

- View list of all Rides
  Generates a report of all the information on Rides.
  Input: List Rides
- View list of all Shops
  Generates a report of all the information on Shops.
  Input: List Shops
- View list of all Mess
  Generates a report of all the information on Mess.
  Input: List Mess

3.2 Employee Interfaces

These functions are available to employees.

- View list of all Employees
  Generates a report of all the information on Employees.
  Input: List Employees
- Modify personal details like, Birthday, Name, Address.
  Input: SSN, New Birthday and/or Name and/or Address
- Modify a Rides entity details like, Cost, VIP.
  Input: Ride ID, New Cost and/or VIP status.
- Modify a Mess entity details like Items.
  Input: Mess Name, New Items
- Modify a Shops entity details like Open.
  Input: Shop License, Open
- Recruit an employee
  Adds an entity under the type Employee.
  Input: First Name, Last Name, SSN, Birthday, Address, Class, Salary.

- Fire an employee
  Removes an entity under the type Employee.
  Input: SSN.

3.3 Visitor/Employee Interfaces

These functions are available to all visitors.

- Issue a Card
  A new card is issued to the visitor with the input details.
  Input: Balance, VIP
- Enjoy Ride
  Deducts the price of the Ride from the Card identified by a Visitor.
  Input: Visitor, Card, Ride, Rating, Review
- Eat Mess
  Deducts the price of the Mess from the Card identified by a Visitor.
  Input: Vistor, Card, Mess.
- Buy from Shop
  Deducts the price of the Shop item from the Card identified by a Visitor.
  Input: Visitor, Card, Shop, Amount