SHENKAR שנקר
ENGINEERING. DESIGN. ART . הנדסה. עיצוב. אמנות
The Pernick Faculty of Engineering . הפקולטה להנדסה ע"ש פרניק
המחלקה להנדסת תוכנה
Software Engineering Dept.

EZDan

A system that makes public transportation commute more accessible for elderly people and for people with disabilities

Software Design Document

Authors:

Suf Karmon ID: 312439474

Adva Shulamit Apelboim ID: 204621486

30/06/2022

Table of Contents

1. Introduction

1.1. System Overview

EZDan is a program made to make public transportation commute more accessible for people with disabilities and elderly people, by using sensors and a highly customizable user interface.

1.2. Problem Description

The public transportation applications on the market today are not suitable for the elderly and for people with disabilities.

- The public transportation applications do not have any accessibility adjustments – this makes people who need those adjustments avoid using the public transportation application, and do not have an efficient way to plan a ride.
- The public transportation applications do not have enough relevant information for those seeking a specific accessible service. The current solutions are suited to the general population and the information they find relevant.
- The right to travel freely is a basic human right, regardless of any characteristics. Public transportation is one of the most basic ways to commute, and therefore should be accessible to everyone. This case can be made by moral ethics, but also by economic motivation to include and service a population of massive scale that is currently lacking the right service.

1.3. Goals

- 40% increase in the use of public transportation by the elderly population (60 years and older).
- 40% increase in the use of public transportation by people with disabilities (carry a disabled permit “Tav Neche”).

1.4. Scope

- Accessibility
- Public transportation application
- IOT and web

1.5. Glossary

- **WCAG** - **Web Content Accessibility Guidelines** - A document stating the manner in which websites and applications should be accessible for all users.
- **Amichay** - **a non-profit association** for the welfare of people with developmental disabilities. Located in Hod-Hasharon, Israel.

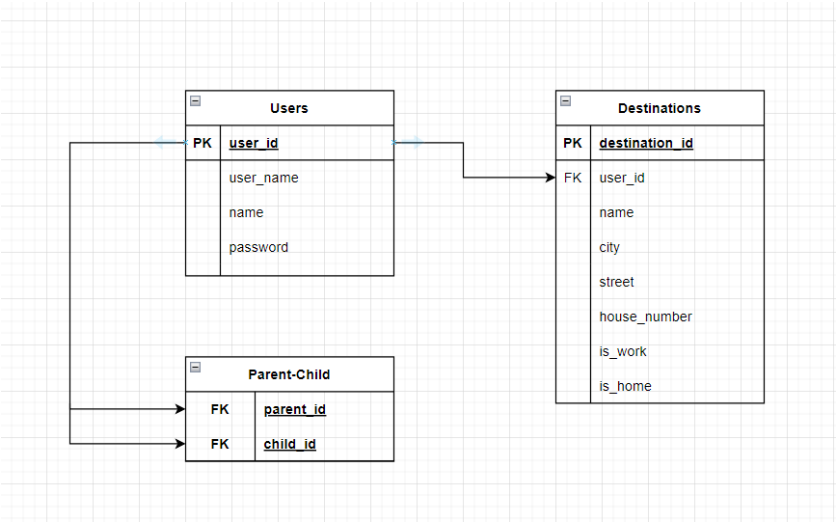
2. System Architecture – System Context Diagram

The system is composed of 2 user types. First is the Parent User which can manage child users. The child user can manage their own destinations and user details. The Parent user can manage each of its child user's destinations and details.

3. System Design

Our system is based on the interaction between a user and it's destinations. We Also have a Ride object which will be stored in a JSON file.

3.1. Data Design - ERD

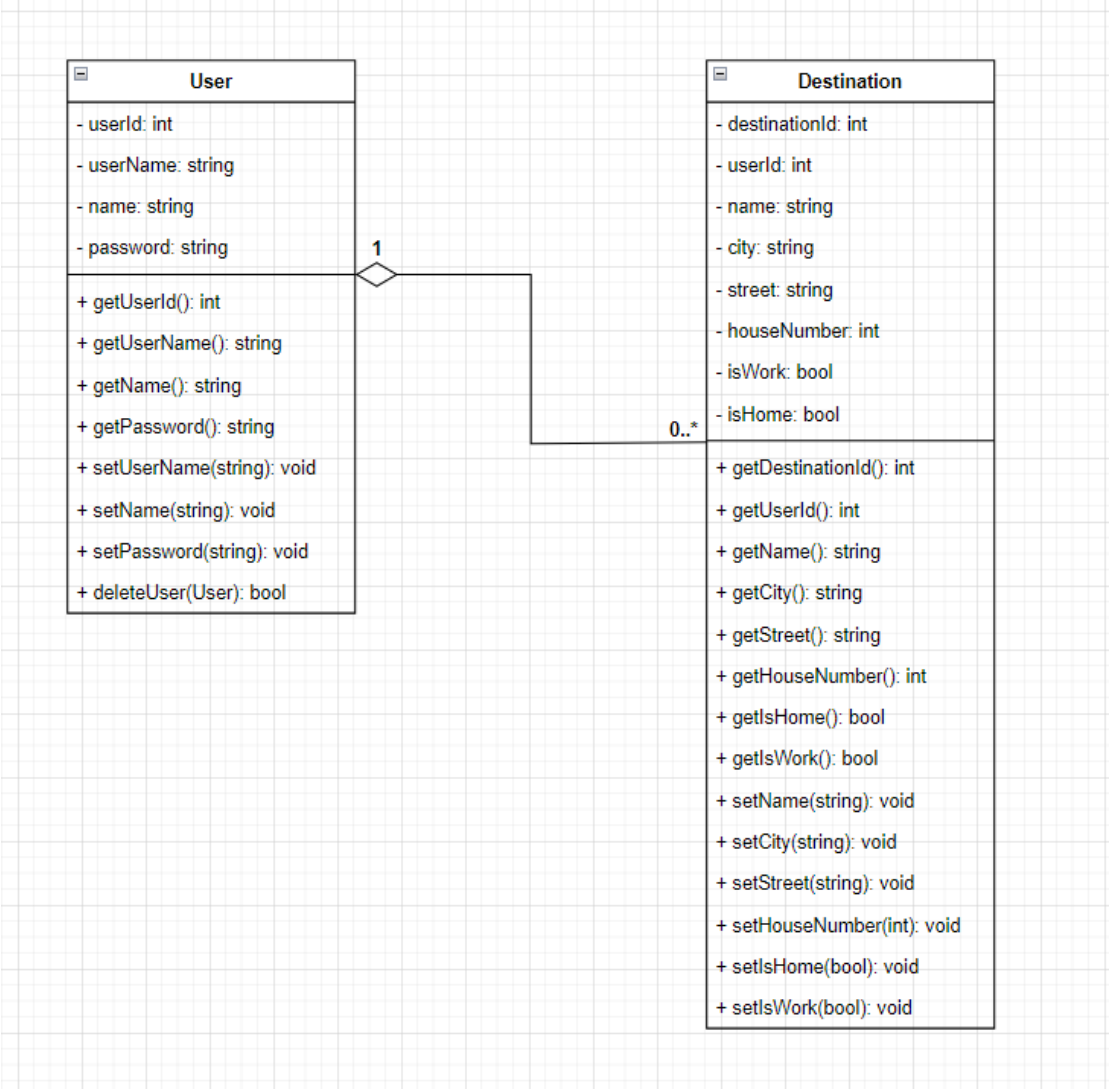


JSON: Ride object - contains all the details regarding the selected ride, including line information and accessibility grades.

```
{
  "id": 123,
  "line": {
    "number": 123,
    "departure": "string",
    "time": "strig"
  },
  "grades": [
    {
      "name": "string",
      "grade": 123,
      "icon": "string"
    },
    {
      "name": "string",
      "grade": 123,
      "icon": "string"
    },
    {
      "name": "string",
      "isAvailable": "boolian",
      "icon": "string"
    },
    {
      "name": "string",
      "availableSeats": 123,
      "icon": "string"
    }
  ]
}
```

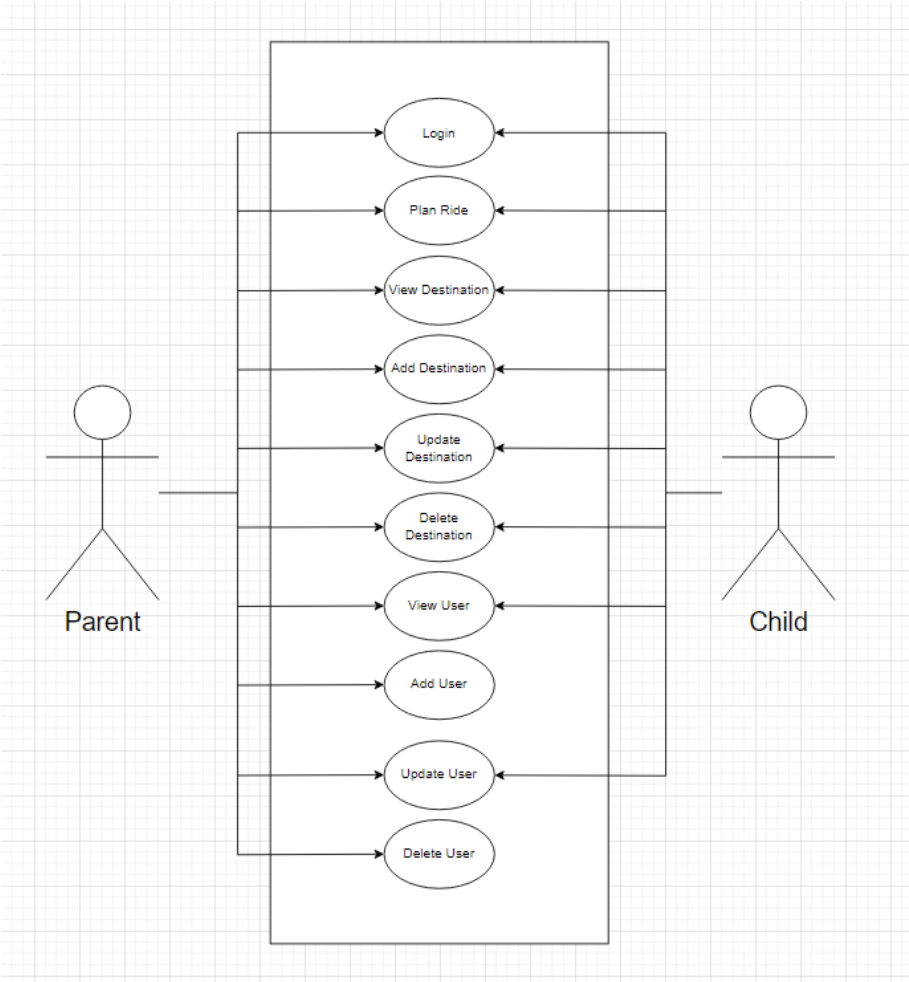
3.2. Structural Design

Class Diagram - every user manages zero to many destinations.



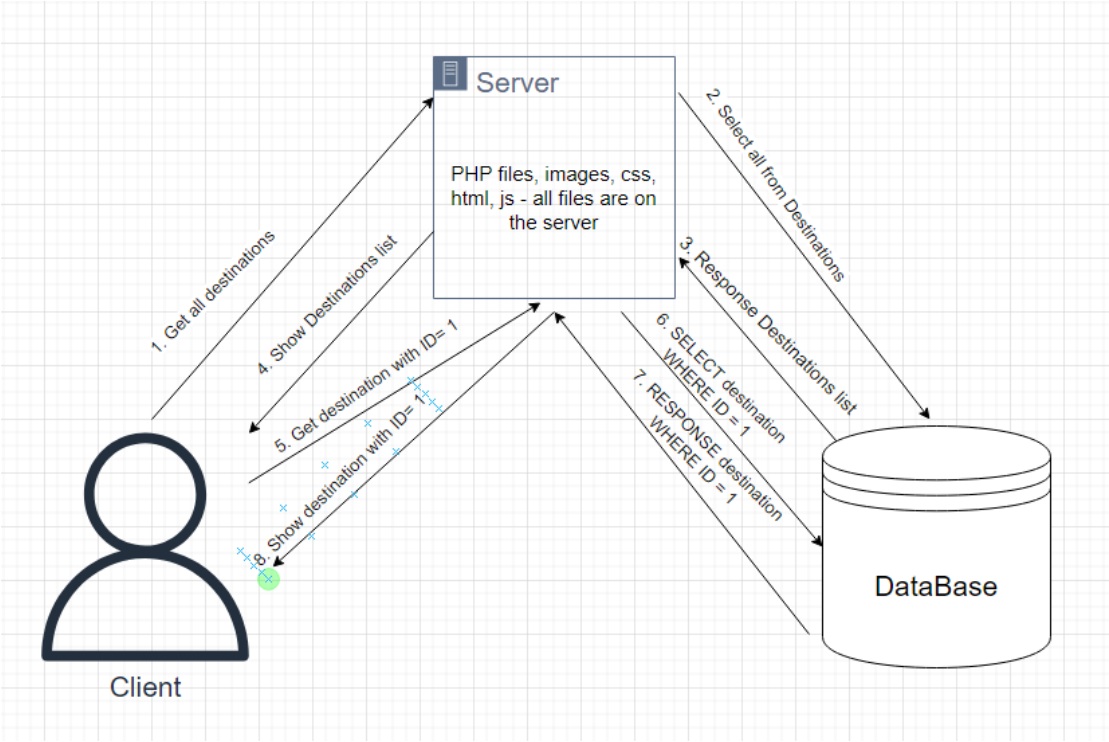
3.3. Interaction Design

Use Case Diagram - parant user can manage both his own details and their child's. A child user can only manage their own details.



4. Software Architecture

The process described in this architecture is getting the destination information from the database.



5. Verification and Validation

In order for the system to work as described in this document, we will have to make the following tests:

- The user can login to the system
  - trying to login with an incorrect password
  - trying to login without filling the form
- The child user can see it's destinations
  - check to see if all destinations are showing
  - check to see that only this user's destinations are shown
- Every user is able to add a destination
- Parent user is able to see it's child users
  - check that child user is not able to see other child users
  - check is the parent user can only see it's own child users
- Every user is able to delete its destinations
  - parent user is also able to delete child destinations
- Parent user can add new child user under him
- every user is able to logout of the system