**Switch Me - A platform for exchanging objects**

Yuval Nurlian 314873209

Sahar Slavkin 206698409

Course: Computer Science Project  
Lecturer: Dr. Sharon Yalov Handzel   
Date:

**1. Introduction**

**Purpose:** To create an intuitive platform for exchanging products between private users in an efficient and fair manner.  
The purpose of the system is to enable users to trade items they no longer need for items they find relevant, leveraging AI-based algorithms for item valuation and matching.

**Overview:** This document describes the detailed design of the exchange platform, including system components, interfaces, data flow, and dependencies.

### 2. System Overview

#### System Structure:

The system consists of three main components: the **Frontend User Interface**, the **Backend API**, and the **Database**.

#### Components and Their Functions:

1. **Frontend User Interface:**
   * Enables users to perform actions such as logging in, uploading items, searching for available items to swap, and managing personal profiles.
2. **Backend API:**
   * Handles requests received from the user interface and processes data.
   * Manages intelligent item matching using AI algorithms.
3. **Database:**
   * Stores user data, uploaded items, swap history, and reviews.

#### Interactions Between Components:

* The **Frontend User Interface** sends requests to the **Backend API**, which processes the requests and returns real-time responses.
* The **Backend API** communicates with the **Database** to store, retrieve, and update data.
* The intelligent matching process is performed by AI algorithms integrated into the **Backend API**, utilizing data from the **Database**.

### 3. Design Considerations

### Assumptions:

* Users will access the system through modern browsers (Chrome, Firefox, Edge) or web-based applications only during the initial phase.
* The system assumes a stable and reliable internet connection for real-time operations.

### Constraints:

* The system will not include mobile application support in the initial phase but will allow for future development if demand arises.
* Third-party AI tools will only support specific models and will require adaptation to the platform.
* The AI mechanism will not be able to estimate all of the items uploaded, but the system aims to achieve the highest possible success rate.
* The database (PostgreSQL) will be designed for medium workloads in the first phase, with scalability for higher workloads in the future.
* Limited development time requires the use of familiar and fast-to-implement technologies such as Angular and Node.js.

**Coding Standards:**

* **Angular, HTML, CSS, TypeScript:** Follow Angular guidelines for structured projects, consistent naming, and readable code.
* **C#:** Adhere to Microsoft C# Coding Conventions.
* **Java:** Use Google Java Style Guide with a focus on OOP principles.
* **SQL:** Write clear queries with consistent table and field names and uppercase keywords.

1. **System Architecture**

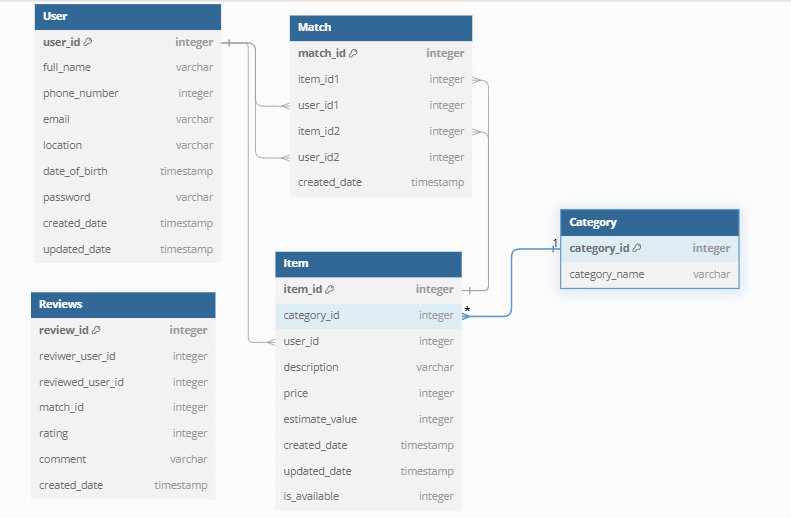
The platform system is based on a client-server architecture. The client (Frontend) enables users to upload products, view other items, and perform exchanges. The server (Backend) processes requests, manages communication with the database, and connects to an external AI API. The database stores all user, product, and exchange process information.

1. **Frontend:**
   * Displays interactive views for users (login, product upload, search).
2. **Backend:**
   * Handles requests, performs authentication, manages business logic, and connects to the AI API.
3. **Database:**
   * Stores user data, item details, and valuation results.
4. **API:**
   * Provides automatic tagging, value estimation, and smart filtering for items.

**5.Unique modules in our system:**

**1. Upload Product Module:**

* **Purpose:** Allows users to upload items to the platform with details such as name, category, description, images, and more.
* **Input:** Item name, category, description, images.
* **Output:** Upload confirmation, unique item ID.
* **Dependencies:**
  + Depends on the Backend for processing and validating input.
  + Depends on the Database to store product details.
  + Depends on the AI API for automatic tagging and value estimation.

**6. Data Design**