Software Requirements Specification (SRS) for myAshkana

1. Introduction

1.1 Purpose

myAshkana is an online food ordering system designed for a university cafeteria. It enables students to pre-order meals and avoid long queues. Additionally, students can vote for new menu items, with the most popular options being introduced weekly.

1.2 Scope

This system will provide:

- Personal accounts linked to myAlatoo.
- An online ordering system with pre-order functionality.
- A weekly voting system for adding new menu items.
- Online and offline payment options.
- Email notifications for order updates.
- Administrative control over menu management and order tracking.

1.3 Definitions, Acronyms, and Abbreviations

- ID-Student – Unique identifier for students used for authentication.
- Pre-order – Ordering food before arriving at the cafeteria.
- Admin – A cafeteria staff member responsible for managing menu items and orders.
- University Email – The official email system used for notifications. (@alatoo.edu.kg)

1.4 References

- myAlatoo API documentation
- Payment gateway API documentation.

---

2. Overall Description

2.1 Product Perspective

myAshkana is a web-based application integrated with the university's authentication system and payment provider. The application follows a role-based access model, allowing students to place orders and cafeteria staff to manage them.

2.2 User Classes and Characteristics

1. Students:

   - Log in using university credentials.
   - Place pre-orders for meals.
   - Must prepay for the first three orders.
   - Vote for new menu items.
   - Cancel orders at least 20 minutes before pick-up.
   - Receive email notifications about order status.

2. Admin (Cafeteria Staff):

   - Manage menu items (add/remove weekly specials).
   - View and process orders.
   - Manage votes and introduce top-voted items into the weekly menu.

3. Cashier:

   - Process offline payments.
   - Confirm order pickup.

2.3 Operating Environment

- Backend: Spring Boot (Java)
- Database: H2 (development), PostgreSQL (production)
- Authentication: University login system (myAlatoo)
- Payment Integration: Online payment gateway
- Notifications: University email system

---

3. Functional Requirements

3.1 Authentication

- The system must authenticate students using their university credentials.
- Unauthorized users must not access ordering functionality.

3.2 Ordering System

- Students can browse the menu and place pre-orders.
- Online prepayment is required for the first three orders before enabling pay-on-pickup.

- The minimum preparation time is 20 minutes from the order submission.

3.3 Menu Management

- The menu consists of permanent items and weekly changing items.
- Admins can update the menu.
- Students can vote for a new menu item each week.
- The top-voted item is added to the menu on a weekly basis.

3.4 Order Management

- Students can cancel an order at least 20 minutes before the scheduled time.
- Orders can be marked as "Ready" and "Picked up" by cafeteria staff.
- Students receive email notifications for order status updates.

3.5 Payment Processing

- Online payment is required for the first three orders.
- After three successful orders, students can opt to pay on pickup.
- Secure payment gateway integration is required.

3.6 Notifications

- Students receive order status updates via university email.
- Admins receive notifications for new orders.

---

4. Non-functional Requirements

4.1 Performance

- Order processing time should not exceed 1 second per request.
- The system should support at least 1000 concurrent users.

4.2 Security

- Authentication must be encrypted and follow university policies.
- Payment transactions must be secure and compliant with industry standards.

4.3 Availability

- The system should be available 99.9% of the time.
- Regular backups must be scheduled to prevent data loss.

---

5. Use Cases

5.1 Student Ordering Flow
1. The student logs in using university credentials.
2. Browses the menu and selects items.
3. Chooses a pickup time.
4. Pays online (if within the first three orders) or selects "Pay on Pickup."
5. Receives a confirmation email.
6. Picks up the order at the cafeteria.

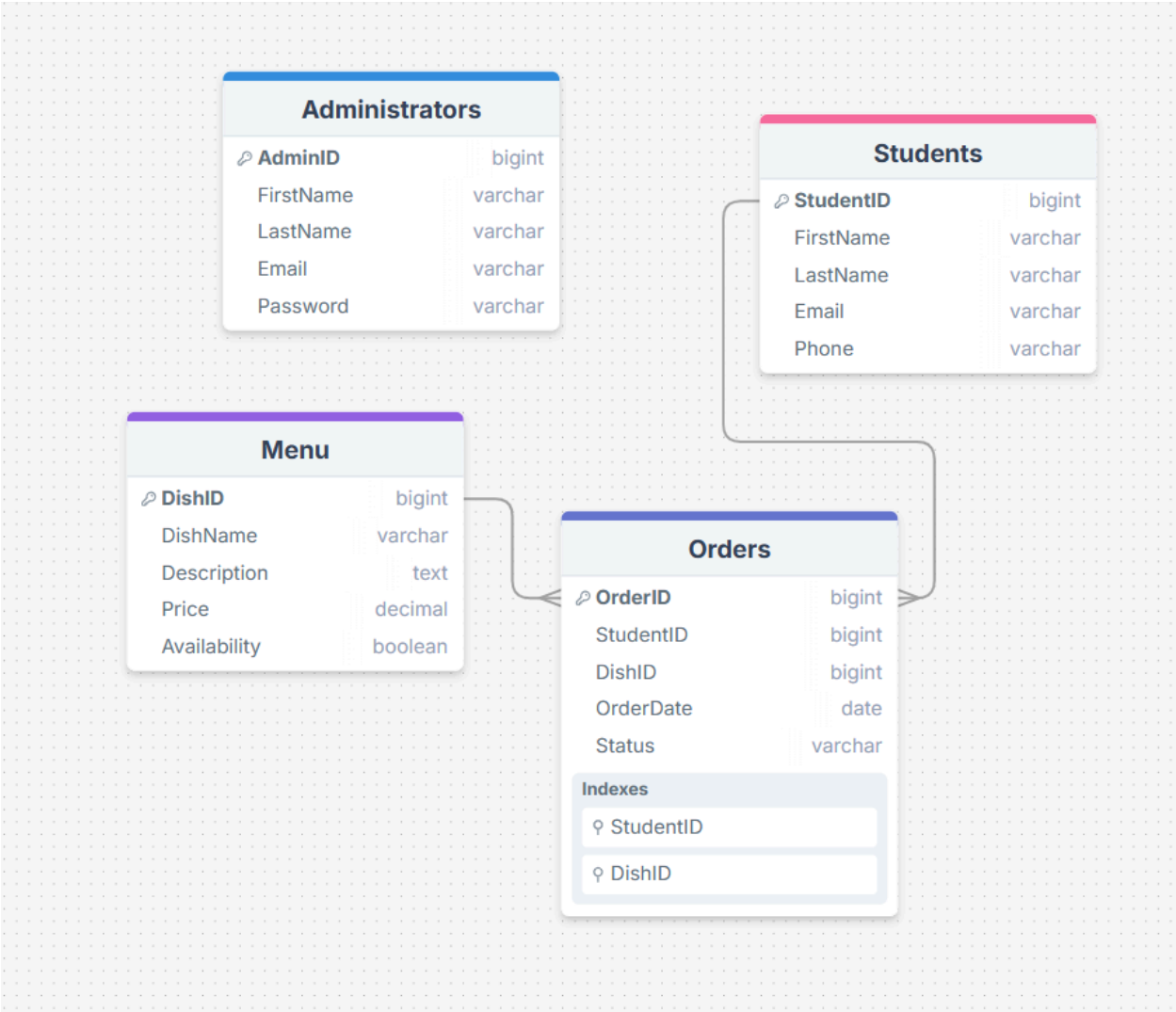
5.2 Admin Menu Management

1. The admin logs in.
2. Views the current menu.
3. Adds or removes items.
4. Confirms the weekly voted item for addition to the menu.

5.3 Voting System

1. Students view proposed new items.
2. Votes are cast weekly.
3. The most voted item is added to the next week's menu.

This document provides a structured approach to developing myAshkana and ensures alignment with the project objectives.


Administrators
AdminID bigint
FirstName varchar
LastName varchar
Email varchar
Password varchar
Students
StudentID bigint
FirstName varchar
LastName varchar
Email varchar
Phone varchar
Menu
DishID bigint
DishName varchar
Description text
Price decimal
Availability boolean
Orders
OrderID bigint
StudentID bigint
DishID bigint
OrderDate date
Status varchar
Indexes
StudentID
DishID