Birzeit University

Department of Electrical & Computer Engineering

First Summer Semester, 2024/2025

ENCS3130 Linux Laboratory

Python Project – Outpatient Reservation System

The **Outpatient Reservation System** is a lightweight terminal-based application developed using **shell scripting**. It allows patients to book appointments and clinic staff to manage doctor schedules. The system uses simple text files for data storage, making it accessible and educational for students learning file handling, menu systems, and input validation in bash.

Project Objectives

- Manage outpatient appointments via a simple text-based interface.
- Allow patients to register, view, and cancel bookings.
- Enable admins to manage doctor information and schedules.
- Prevent double-booking and validate user inputs.
- Provide system backup and logging functionality.

Login System

Users select their role upon launching the application:

- Patient – can register, view, or cancel appointments
- Admin – can manage doctors and schedules, view reports, and perform backups

File Structure

The system stores all data in three pipe-delimited text files:

1. doctors.txt – Doctor Info

   Format: DoctorID|Name|Specialty|AvailableDays|StartTime|EndTime

2. patients.txt – Patient Info

   Format: PatientID|Name|Phone

3. appointments.txt – Appointment Info

   Format: AppointmentID|PatientID|DoctorID|Date|Time|Status

Core Functionalities

Patient Services

- **Register a New Patient**
  Allows new users to sign up by entering their name and phone number. A unique Patient ID is automatically generated and stored in the system's `patients.txt` file for future appointment management.
- **View Appointment History**
  Displays all appointments (past and upcoming) associated with the logged-in patient. This includes details such as the doctor's name, specialty, appointment date and time, and current status (e.g., Confirmed, Cancelled).
- **Cancel Appointments**
  Enables patients to cancel upcoming appointments. Instead of removing the entry, the system updates its status to "Cancelled," preserving a complete historical record.

Admin Services

- **Add a New Doctor**
  Lets the administrator register a new doctor by providing their name, medical specialty, available working days, and working hours. The information is saved in `doctors.txt` for scheduling and lookup purposes.
- **View Doctor's Schedule**
  Displays all booked appointment slots for a selected doctor, along with patient names and appointment times. This helps administrators track doctor availability and manage scheduling efficiently.
- **Backup the System Data**
  Initiates a manual backup process, duplicating all core data files into a dedicated `backups/` directory. This ensures data recovery options in case of accidental loss or corruption.

Additional Features

- **Backup**
  The system includes a built-in feature for administrators to create backups of the data files (`doctors.txt`, `patients.txt`, and `appointments.txt`). These are copied into a separate `backups/` folder to protect against accidental data loss and support system recovery.
- **Logging**
  All critical operations—such as patient registration, doctor addition, appointment bookings and cancellations, and backup creation—are recorded in a log file (`system.log`). Each entry is timestamped, helping maintain an audit trail and aiding in troubleshooting or user activity analysis. Each log entry is a single line containing a **timestamp**, **action type**, and **action details**.

Key Features

- **Menu-Based Interface**
  The system uses a simple, intuitive text menu that presents numbered options to users. Both patients and admins can navigate easily without requiring technical skills.
- **Role-Based Access**
  Access to features is based on user roles. Patients can only access personal booking functions, while administrators have additional permissions to manage system-wide data and operations.
- **Data Validation**
  - Ensures phone numbers and IDs are correctly formatted and numeric.
  - Validates input dates follow the standard format `YYYY-MM-DD`.
  - Prevents duplicate entries for patients and doctors based on existing IDs.
  - Checks whether essential data files exist before performing read/write operations.
- **Conflict Detection**
  - Automatically detects and prevents double-booking of appointment slots for both patients and doctors.
  - Ensures appointments are only scheduled on valid days and within the working hours defined for each doctor.

Submission Instructions

Submit a compressed .zip folder containing:

- Source code, files, ...
- A README.md file explaining code usage, execution instructions, and system features.

Notes:

- Write the code for the shell program to satisfy the requirements described above.
- Make sure your code is clean and well indented; variables have meaningful names, etc.
- Make sure your code has enough comments inserted to add clarity.
- Students must work in groups of no more than two. Working alone is not permitted and will result in a 30% deduction from the final project grade.
- Deadline: **Friday,** 15 **August, 2025 at 11:59pm.** Please submit your project (code + test cases) through Ritaj as a reply to this message.
- This project is per group effort: instances of cheating will result in you failing the course.