Comprehensive Exercise Report

Team Saratha\_Hasaru of Section ADBDW

Nauttuduwa Liyanage Don Hasaru Hiruditha 223AEB022

Saratha.Widhana 233AEB017

NOTE: You will replace all placeholders that are given in <<>>

[**Requirements/Analysis**](#_uwgqwd5ezv2w) **2**

[Journal](#_lsityg2iq9m6) 2

[Software Requirements](#_2h0vru1u2mla) 3

[**Black-Box Testing**](#_prhaxdxmf8n8) **4**

[Journal](#_18f11w613jft) 4

[Black-box Test Cases](#_2xn4jzot820y) 5

[**Design**](#_24fdizefyocn) **6**

[Journal](#_esp2ocs9j6bk) 6

[Software Design](#_aifbl1x6rddt) 7

[**Implementation**](#_hya8f3jqkba6) **8**

[Journal](#_acupzfhai7gz) 8

[Implementation Details](#_ojhtwkms2z3b) 9

[**Testing**](#_3qvya3vi836q) **10**

[Journal](#_ckfs4xbl5pyr) 10

[Testing Details](#_bzt1547yxzxi) 11

[**Presentation**](#_hdjvrbf45b1p) **12**

[Preparation](#_xbiquwtmf36n) 12

[**Grading Rubric**](#_u0hfnmdgusmf) **13**

# Requirements/Analysis

Week 2

## Journal

The following prompts are meant to aid your thought process as you complete the requirements/analysis portion of this exercise. Please respond to each of the prompts below and feel free to add additional notes.

* After reading the client’s brief (possibly incomplete description), write one sentence that describes the project (expected software) and list the already known requirements.
  + web-based online booking system designed for a barbershop, enabling customers to schedule appointments and barbers to manage their schedules efficiently.
    - Customers can register, log in, and manage their profiles.
    - Customers can view available barbers and book appointment slots.
    - Appointment confirmation and cancellation options are available.
    - Barber (admin) can add, edit, and remove time slots.
    - Barber can view and manage all bookings from a dashboard.
* After reading the client’s brief (possibly incomplete description), what questions do you have for the client? Are there any pieces that are unclear? After you have a list of questions, raise your hand and ask the client (your instructor) the questions; make sure to document his/her answers.
  + What specific services should be listed (e.g., haircut, beard trim, hair coloring)?
  + What are the shop's working hours and days off?
  + How long is each type of service or appointment?
  + Do you have any logo, color scheme, or design preferences?
* Does the project cover topics you are unfamiliar with? If so, look up the topics and list your references.
  + No
* Describe the users of this software (e.g., small child, high school teacher who is taking attendance).
  + Individuals who want to book hair-related services online.(**especially focus on international students in Riga**)
  + Barbers working at the barbershop.l
* Describe how each user would interact with the software
  + Customers interact with the system by registering an account and logging in to book appointments for services such as haircuts or beard trims.
  + Barbers log into the system to view their daily schedules, check upcoming appointments, and manage their availability
* What features must the software have? What should the users be able to do?

Customers can

* + Register and log in to their account securely.
  + View available services and prices.
  + View available dates and time slots.
  + Book an appointment.

Barber can

* + View personal schedule of upcoming appointments.
  + Update their availability (working hours or days off).
  + See customer details and booked services.
* Other notes:
  + The system is designed as a web-based application, making it accessible from any device with internet access, including smartphones, tablets, and computers.

## Software Requirements

This project is a web-based booking system developed for a local barbershop to streamline the appointment scheduling process for both customers and barbers. The system allows customers to register, log in, browse services, and book appointments based on real-time availability. It also provides barbers with a personalized view of their schedule and the ability to manage their availability. The admin (typically the shop owner) has full control over the platform, with the ability to manage bookings,services, and customers through an intuitive dashboard. The software emphasizes ease of use, data security, and responsiveness across devices, ensuring a smooth experience for all user types.

**System Requirements**

* The system shall allow customers to register and log in securely.
* The system shall allow customers to view a list of services with descriptions and prices.
* The system shall allow customers to book appointments
* The system shall send confirmation emails upon successful bookings.
* The system shows baber personal contact details on website.
* The system shall log and store booking data securely in a database.

|  |
| --- |
|  |

**Customer User Stories**

a customer said, I want to view my booking history so that I can keep track of past services and appointments.

a customer said, I want to receive friendly reminders before my appointment so I’m less likely to miss it.

a customer said , I want to receive friendly reminders before my appointment so I’m less likely to miss it.

a customer said , we need more barbers in your shop then we can choose as we prefer.

# Black-Box Testing

Instructions: Week 4

## Journal

***Remember:*** Black box tests should only be based on your requirements and should work independent of design.

The following prompts are meant to aid your thought process as you complete the black box testing portion of this exercise. Please review your list of requirements and respond to each of the prompts below. Feel free to add additional notes.

* What does input for the software look like (e.g., what type of data, how many pieces of data)?

Customer Registration or Login

* + Name (text)
  + Email address (text, must be in valid format)
  + Password (text)

Booking an Appointment

* + - Selected service (from a dropdown list)
    - Preferred date (date picker)
    - Preferred time slot

Admin Input

* + - New service name and price (text + number)
    - Modifications to barber availability (date/time selections)
    - Booking overrides or deletions (selection-based)
* What does output for the software look like (e.g., what type of data, how many pieces of data)?

After Booking an Appointment:

* + - Appointment details: service, date, time

Login**:**

* + - Dashboard or schedule view showing:
    - Upcoming appointments
    - Appointment date and time
* What equivalence classes can the input be broken into?

1. Login Credentials

Valid Class:

* + - Properly formatted email and correct password

Invalid Classes:

* + - Incorrect email format (e.g., user@)
    - Incorrect password
    - Blank email or password fields

1. Appointment Booking

Valid Class:

* + - * Selected service, available time slot, and future date

Invalid Classes:

* + - * No service selected
      * Time slot already booked
      * Past date selected
* What boundary values exist for the input?

Password Length (Registration/Login)

* + - Minimum accepted length: 8 characters
    - Emails must follow a specific format ([name@gmail.com](mailto:name@gmail.com))

Booking Date

* + - Must be today or in the future
* Are there other cases that must be tested to test all requirements?
  + **S**ecurity and Authentication Cases
  + Booking Edge Cases
  + Data Handling & Feedback
* Other notes:

## Black-box Test Cases

Use your notes from above to complete the black-box test plan section of the formal documentation by writing black box test cases (other than actual results since no program currently exists). Remember to test each equivalence class, boundary value, and requirement.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Description** | **Expected Results** | **Actual Results** |
| Test 1 | Register with all valid inputs | Account is created successfully ,user is redirected to login | Account is created successfully |
| Test 2 | Try booking an apoinment in a past date | System rejects input with error “cannot book in the past” | System rejects input with error |
| Test 3 | Book two apoinment at the same time slot | Second booking is rejected with error “slot unvailable” | Second booking is rejected |
| Test 4 | Enter a password with exactly 8 characters during registration | System accepts input ;account is created succesfully | Account created successfully |
| Test 5 | Admin tries to create a service with negative price | System rejects input ;error message “price must be a positive number”is shown | System rejects input |
|  |  |  |  |
|  |  |  |  |

# Design

Instructions: Week 6

## Journal

***Remember:*** You still will not be writing code at this point in the process.

The following prompts are meant to aid your thought process as you complete the design portion of this exercise. Please respond to each of the prompts below and feel free to add additional notes.

* List the nouns from your requirements/analysis documentation.
  + User
  + Customer
  + Admin
  + Barber
  + Appointment
  + Service
  + Booking
  + Date
  + Time
  + Email
  + Password
* Which nouns potentially may represent a class in your design?
  + User – base class for customers and admins (attributes: name, email, password)
  + Customer – subclass of User (can book appointments)
  + Admin – subclass of User (can manage services and view bookings)
  + Booking
  + Timeslot
* Which nouns potentially may represent attributes/fields in your design? Also list the class each attribute/field would be a part of.
  + Name-
  + Email-
  + phoneNumber-
  + date-
* Now that you have a list of possible classes, consider different design options (***lists of classes and attributes***) along with the pros and cons of each. We often do not come up with the best design on our first attempt. Also consider whether any needed classes are missing. These two design options should not be GUI vs. non-GUI; instead you need to include the classes and attributes for each design. Reminder: Each design must include at least two classes that define object types.
* Which design do you plan to use? Explain why you have chosen this design.
  + We are planning to use an Object-Oriented Design for our barbershop booking system. This design approach allowed us to model real-world entities—such as users, appointments, and time slots—as software objects with properties and behaviors.
* List the verbs from your requirements/analysis documentation.
  + Register
  + Login
  + Logout
  + Book
  + Select
* Which verbs potentially may represent a method in your design? Also list the class each method would be part of.
  + register()
  + login()
  + logout()
  + updateProfile()
  + viewAppointments()
* Other notes:

## Software Design

<<Use your notes from above to complete this section of the formal documentation by planning the classes, methods, and fields that will used in the software. Your design should include UML class diagrams along with method headers. ***Prior to starting the formal documentation, you should show your answers to the above prompts to your instructor.****>>*

**Classes and Methods;**

**User:**

**Fields:**

id: int(11)

name:varchar(100)   
email: varchar(100)

password: varchar(100)

contact: varchar(100)

Methods:

login(email: string, password: string): boolean

logout(): void

updateProfile(name: string, phoneNumber: string): void

Appoinments

Fields:

- id: int

- userId: int

- slot\_id:int

- bookingtime: dateTime

-

Methods:

+ confirm(): void

+ cancel(): void

+ reschedule(newDate: Date, newTime: Time): void

**A screenshot of a computer

AI-generated content may be incorrect.**

# Implementation

Instructions: Week 8

## Journal

The following prompts are meant to aid your thought process as you complete the implementation portion of this exercise. Please respond to each of the prompt below and feel free to add additional notes.

* What programming concepts from the course will you need to implement your design? Briefly explain how each will be used during implementation.
  + Object-Oriented Programming (OOP)- The system is built around real-world entities like User, Appointment, and TimeSlot. Each of these is implemented as a class with attributes (fields) and methods. Inheritance is used to extend base classes like User.

We created classes to represent the key entities in the system:

* + User: Represents customers who can register, log in, and book appointments.
  + Appointment: Represents a scheduled service with a date and time slot.
  + TimeSlot: Represents available times that users can select for appointments.
* Other notes:

## Implementation Details

<<Use your notes from above to write code and complete this section of the formal documentation with a README for the user that explains how he/she will interact with the system.>>

Basic Requirments

-web browser to use the website  
  
Who Is This For?

-Customers who want to book haircuts or grooming services online without calling or visiting in person.

How to Use the System

-Go to the website home page.

-Click Sign Up to register as a new user by entering:

* + Name
  + Email
  + Phone Number
  + Password

-Once registered, use Log In with your email and password to access the dashboard.

-After logging in, click on Book Appointment.

-Choose:

* A Date from the calendar.
* An available Time Slot (shown as enabled).

- Click Confirm Booking. Your appointment will be saved and shown in your dashboard.

# Testing

Instructions: Week 10

## Journal

The following prompts are meant to aid your thought process as you complete the testing portion of this exercise. Please respond to each of the prompts below and feel free to add additional notes.

* Have you changed any requirements since you completed the black box test plan? If so, list changes below and update your black-box test plan appropriately.
  + Nothing changed
* List the classes of your implementation. For each class, list equivalence classes, boundary values, and paths through code that you should test.

-User  
Equivalence classes- Valid credentials

Boundary values- Password length

Paths through code- register()>success<>email exists<>invalid input

-Appointment

Equivalence classes-Future date & free slot

Boundary values-Date ,today and future

Paths through code- cancel()>normal cancel> already cancelled/completed

* Other notes:

## 

## 

## Testing Details

<<Use your notes from above to write your test programs and complete this section of the formal documentation by creating a list of your test programs along with descriptions of what they are testing. You will also complete the black-box test plan by running the program and filling in the Actual Results column.>>

test\_user\_registration()-Tests user registration with valid and invalid inputs (e.g., missing fields, short passwords, duplicate emails)

test\_user\_login()-Verifies login for correct credentials, wrong password, and non-existent email

test\_appointment\_booking()-Attempts to book appointments with valid data, a past date, or an already-booked slot

test\_timeslot\_availability()-Tests whether a time slot is available before and after booking

# Presentation

Instructions:Week 12

## Preparation

The following prompts are meant to aid your thought process as you complete the presentation portion of this exercise. It is recommended that you examine the previous sections of the journal and your reflections as you work on the presentation as it is likely that you have already answered some of the following prompts elsewhere. Please respond to each of the prompts below and feel free to add additional notes.

* Give a brief description of your final project
  + Our final project is an online barbershop booking system called **Eazy Barber**. It is a web-based platform that allows customers to register, log in, and book appointments with a barber. The system also provides the barber (admin) with the ability to view appointments and manage time slots. The goal of the system is to eliminate manual appointment handling and reduce scheduling conflicts.
* Describe your requirement assumptions/additions.
  + We assumed that the system would have only **one barber**, so there was no need to implement multi-barber assignment logic. We added additional requirements such as form validation, mobile responsiveness, and email confirmation for booking success to improve the user experience. We also included an admin dashboard feature for the barber to view and manage appointments.
* Describe your design options and decision. How did you weigh the pros and cons of the different designs to make your decision?
  + We considered both procedural and object-oriented designs. After evaluating scalability, modularity, and maintainability, we chose **object-oriented programming**. This allowed us to separate responsibilities between classes like User, Appointment, and TimeSlot. It also made the system easier to expand later (e.g., for multiple barbers ).
* How did the extension affect your design?
  + <<Insert answer>>
* Describe your tests (e.g., what you tested, equivalence classes).

We performed black-box testing based on functional requirements. We tested:

* + Registration/login with valid and invalid inputs (equivalence classes)
  + Boundary values like minimum password length and time slot limits
  + Booking in the past (invalid), double booking (invalid), and normal bookings (valid)
* What lessons did you learn from the comprehensive exercise (i.e., programming concepts, software process)?
  + We learned how to apply object-oriented principles to real-world problems, especially the use of encapsulation, inheritance, and method abstraction. We also gained experience in planning the software development life cycle from requirements gathering to testing and documentation
* What functionalities are you going to demo?
  + User registration and login
  + Booking an appointment by choosing a service, date, and time slot
  + Admin view of all scheduled appointments
  + Real-time time slot availability based on bookings
* Who is going to speak about each portion of your presentation? (Recall: Each group will have ten minutes to present their work; minimum length of group presentation is seven minutes. Each student must present for at least two minutes of the presentation.)
* Other notes: