RESERVİFY

Course: Project Management

Team Members: Kağan Erdem, Ahmet Hilmi Büber, Buğra Sarıkaya

Submission Date: May 2025

Github: https://github.com/Kgnerdm/RESERVIFY

**Introduction**

Reservify is a modern, web-based appointment and reservation management platform designed to meet the needs of small and medium-sized businesses such as barbershops, dental clinics, gyms, and personal services. The platform aims to simplify and automate the booking process, providing users with an intuitive experience while giving businesses powerful tools to manage and organize their schedules efficiently.  
  
The project focuses on core features like customer appointment booking, business profile management, administrative dashboards, email notification systems, conflict management to avoid double bookings, and a scalable cloud-based deployment architecture.

# Table of Contents

1. Project Plan  
2. Task Assignment  
3. Document List  
4. Progress Tracking Table  
5. Team Policies and Conflict Resolution

# 1. Project Plan

## Objective

The objective of this project is to design and develop a professional web-based platform that allows businesses such as barbershops, dental clinics, and gyms to efficiently manage customer appointments. The system will offer a user-friendly interface for both clients and businesses, streamlining appointment scheduling, tracking, and management.

## Scope

- Business Registration and Profile Management  
- Customer Appointment Booking  
- Admin Dashboard for Appointment Management  
- Email Notification System  
- Conflict Management (Avoid double-booking)  
- Security and User Authentication  
- Cloud Deployment

## Deliverables

- Fully functional web platform with responsive design  
- Admin and user dashboards  
- Structured database schema  
- Cloud-based deployment and hosting  
- User manuals and technical documentation

## Timeline

Total Project Duration: 4 months

|  |  |  |
| --- | --- | --- |
| Phase | Duration | Description |
| Requirements Gathering | 2 weeks | Collecting business needs and defining user requirements |
| System Design and Architecture | 2 weeks | Creating wireframes, database schema, and workflows |
| Development Phase 1 (Core Features) | 4 weeks | Building registration, booking, and dashboards |
| Development Phase 2 (Enhancements) | 4 weeks | Email notifications, conflict management, UI/UX improvements |
| Testing and Quality Assurance | 2 weeks | Unit testing, integration testing, load testing |
| Deployment and Launch Preparation | 1 week | Deployment to cloud server and final adjustments |
| Project Presentation and Final Review | 1 week | User manual, project documentation, and presentation |

## Risks and Mitigation Strategies

|  |  |
| --- | --- |
| Risk | Mitigation |
| Server downtime during deployment | Use reputable cloud services with backup strategies |
| Task delays due to unforeseen circumstances | Weekly progress meetings and reassignments if necessary |
| Miscommunication among team members | Adhere strictly to team communication policies |
| Low user adoption rates | Prioritize intuitive and user-friendly design practices |

# 2. Task Assignment

|  |  |
| --- | --- |
| Task | Assigned To |
| Gather project requirements | Kağan Erdem |
| Analyze user needs | Kağan Erdem |
| Create main user flow diagram | Kağan Erdem |
| Create basic website sitemap | Kağan Erdem |
| Plan admin login design | Kağan Erdem |
| Define reservation form requirements | Kağan Erdem |
| Design database schema | Kağan Erdem |
| Prepare homepage prototype | Kağan Erdem |
| Create GitHub repository | Kağan Erdem |
| Open and manage GitHub issues | Kağan Erdem |
| Assign GitHub tasks | Kağan Erdem |
| Integrate modules and initial testing | Kağan Erdem |
| List service categories | Ahmet Hilmi Büber |
| Define business registration form | Ahmet Hilmi Büber |
| Design business profile page | Ahmet Hilmi Büber |
| Design reservation calendar | Ahmet Hilmi Büber |
| Design reservation history page | Ahmet Hilmi Büber |
| Plan email notification system | Ahmet Hilmi Büber |
| Backend for reservation times | Ahmet Hilmi Büber |
| Code admin reservation management | Ahmet Hilmi Büber |
| Implement email notifications | Ahmet Hilmi Büber |
| Business profile update function | Ahmet Hilmi Büber |
| Reservation cancel/update system | Ahmet Hilmi Büber |
| Module testing | Ahmet Hilmi Büber |
| Research server and domain options | Buğra Sarıkaya |
| Prepare deployment plan | Buğra Sarıkaya |
| Create testing scenarios | Buğra Sarıkaya |
| Support frontend development | Buğra Sarıkaya |
| Develop conflict check system | Buğra Sarıkaya |
| Server-side error handling | Buğra Sarıkaya |
| User reservation module | Buğra Sarıkaya |
| User login/registration | Buğra Sarıkaya |
| Reservation cancel module | Buğra Sarıkaya |
| Project progress report | Buğra Sarıkaya |
| Conduct load testing | Buğra Sarıkaya |
| Finalize project demo | Buğra Sarıkaya |

# 3. Document List

Kağan Erdem:  
- Project Requirements Analysis Document  
- User Flow Diagram  
- Website Sitemap Plan  
- GitHub Repository Management Report  
- Test Report

Ahmet Hilmi Büber:  
- Business Needs Analysis  
- Admin Panel Design Document  
- Reservation Calendar Design  
- Email Notification System Documentation  
- Test Report

Buğra Sarıkaya:  
- Server/Domain Selection Report  
- Deployment Plan  
- Test Scenario Document  
- User Module Technical Documentation  
- Load Testing Report

# 4. Progress Tracking Table

The detailed task progress tracking table is available in the attached Excel file.

# 5. Team Policies and Conflict Resolution

**Team Policies and Conflict Resolution Strategy**

**1. Communication and Collaboration Policy**

* All team members are required to participate in scheduled weekly meetings and provide status updates.
* Major announcements, task updates, and changes must be shared on the team’s official communication channel within 12 hours.
* Active communication is mandatory. All team members must respond to project-related inquiries within 24 hours.

**2. Task Management and Accountability**

* Each task must be clearly defined, assigned, and tracked with specific deadlines using GitHub Issues.
* If a member anticipates a delay or encounters an obstacle, they must inform the team at least 48 hours before the task deadline.
* Repeated failure to meet deadlines without valid justification will result in task reallocation and official team review of responsibilities.

**3. Conflict Resolution Procedure**

**Step 1 – Direct Negotiation:**

* Team members involved in a disagreement must attempt to resolve the issue through respectful and direct discussion within 24 hours of the conflict arising.

**Step 2 – Team Mediation:**

* If direct negotiation fails, the issue must be escalated to a full team meeting, where the team collectively discusses and proposes solutions.

**Step 3 – Democratic Voting:**

* If mediation does not resolve the conflict, the team will hold an anonymous majority vote (2 out of 3 votes) to determine the resolution.

**Step 4 – Supervisor Notification (if applicable):**

* In case of an unresolved conflict impacting deliverables, the issue will be documented and, if necessary, reported to the course instructor for further guidance.

**4. Quality Assurance and Review Policy**

* Every completed task must undergo peer review by at least one other team member before it is considered finalized.
* Full project review sessions will be conducted 48 hours prior to major deadlines to ensure quality, consistency, and adherence to project requirements.

**5. Risk Management and Contingency Planning**

* In the event that a team member is unable to continue (due to illness, emergencies, etc.), their assigned tasks will be promptly reassigned among available members.
* Backup plans for critical tasks will be identified early during the planning phase to minimize project disruptions.