Online Food Delivery Service Software Development

Table of contents

1. Introduction

The project's goal is to create an online food delivery service's user-friendly platform. This platform will assist with organizing deliveries, payments, costs, and client and employee accounts. The system's actors include clients, managers, banks, restaurants, employees, and map service.

2. Requirements

Actors (any person or system that interacts with your system):

- **Client**
  - Register
  - Make a request
  - Make an appointment to receive the request
  - Payment
  - Reclamation
  - Delete the request
  - Remove the account
  - Leave a review on the restaurant
  - Leave a review on the employee

- **Managers (people working for the company)**
  - Manage the dbs
  - Create the account for the restaurant
  - Verify the restaurant account
  - Create the account for the delivery (employee)
  - Verify the delivery account
  - Publicity
  - Coordinate with banks

- **Banks**
  - Handle payments

- **Restaurants**
  - Create the account
  - Manage menu items and availability

- **Employees (transportist, deliverer)**
    - Receive a message to serve a request (receive the client information, …)
    - Track the delivery in a map
    - Leave the job
    - Leave a complaint on the restaurant
    - Create the account
    - Delete account
    - Online chat with the client
    - Send an alert to the client

- **Map service**
    - Optimize the delivery route
    - Display the route to the employee and client

3. Tasks

- **Analysis and Requirements Gathering**
    - Understand project goals.
    - Get all the system requirements.
    - Gather and document system requirements.

- **System Design**
    - Design software architecture.
    - Define database schema.
    - Establish system interfaces.
    - Create user flow diagrams.

- **Implementation**
    - Develop client components
    - Develop server components
    - Implement client-side logic.
    - Implement a method to do payments

- **Testing**
    - Unit testing for individual components.
    - Integration testing for the entire system.
    - Stress testing for evaluating system performance under high load.

- **Deployment**
    - Set up servers
    - Deploy the application
    - Ensure operational stability

- **Maintenance**
    - Continuous monitoring for errors
    - Security vulnerabilities; periodic updates based on user feedback.

4. Work Schedule

The estimated time for every task listed before should be displayed in this section. I made an attempt to follow the plan for the prior exercise, but my estimations of the timeframes are not precise, it is an estimation.

Task Timeline:

- **Analysis and Requirements Gathering**
  - 2 weeks, 2 people
- **System Architecture Design**
  - 3 weeks, 2 people
- **Implementation**
  - 4 months, 5 people
- **Testing**
  - 1 month, 3 people
- **Deployment**
  - 3 weeks, 2 people
- **Maintenance**
  - Ongoing, 2 people