

# C# 2 Project Overview

The Recruitment Tracking System is designed to streamline recruitment processes for temporary staffing agencies (uitzendbureaus) and employment agencies. This system offers features to effectively manage job advertisements, organise applicant data, handle employer relationships, and coordinate applications seamlessly.

These agencies typically manage a large volume of job openings and applications, so having efficient tools that allow them to monitor recruitment activities and refine hiring strategies is critical.

## Target Users

The primary users of this application are employment agents, these are specialists who require tools to manage job placements and monitor applicants' progress through various stages of recruitment. This system is designed to make complex tasks as simple as possible so agents can track and manage multiple candidates and job postings efficiently.

In addition this application could be very useful for temp hiring agencies. These agencies frequently handle a high volume of job openings and applicants. These agents need to process these applicants and jobs rapidly and efficiently, so they need a tool that allows them to handle numerous applications quickly and seamlessly.

## Core Functionalities

### Job Listing Management

- **Create and post job vacancies:** Users can easily create job postings with detailed descriptions, requirements, and qualifications. This includes setting terms such as duration for temporary positions or specifying full-time or part-time status.
- **Edit and update job vacancies:** Provides flexibility to modify job postings as requirements change or to correct errors.
- **Archive/delete job vacancies:** Enables users to remove outdated or filled positions to keep the job board current and relevant.

### Applicant Data Organization

- **Applicant profiles:** Securely store and manage comprehensive applicant profiles that include resumes, contact information, cover letters, and additional documents.
- **Applicant tracking:** Monitor and update the status of applicants as they move through different stages of the hiring process, from initial application to final decision.

- **Search applicants:** Search capabilities allow users to filter applicants based on specific criteria such as experience level, education, skills, or application status.

## Employer Management

**Employer profiles:** Maintain detailed profiles for each employer, including company overview, contact details, and preferences for candidate qualifications.

## Application Tracking

Enhance the application process by providing tools to track each stage of the application, such as meetings, evaluations, comments, updates and information about the application

## Reporting and Analytics

Generate insightful reports on recruitment activities, performance metrics, and outcome analyses to aid strategic planning and decision-making.

# Moscow analysis

## Must Have

- Job posting management
- Applicant tracking
- Application tracking
- Employer profile management
- Database integration
- User authentication

## Should Have

- Advanced search and filtering
- Logging
- Reporting tools
- Interactive dashboard
- Bulk actions for management

## Could Have

- Automated communication tools

## Won't Have

- Integration with external job boards such as linkedin
- Complex analytics
- Extensive customization for end users

- Integration with external tools such as calendars

## System Requirements

### Functional Requirements

- User authentication
- Data entry
- Data storage and retrieval
- Search functionality

### Non-Functional Requirements

- Usability
- Scalability
- Performance (e.g., load times should be minimal)
- Security measures for data protection

## System Architecture

**Frontend:** WPF

**Backend:** C# with .NET 6

**Database:** SQL Server Express via SSMS

Framework: .NET 6

.NET 6 was chosen for its long term support and cross platform capabilities, making for a scalable and robust application.

Architecture Type: Model-View-ViewModel (MVVM)

MVVM was chosen because it promotes separation of concerns, which improves maintainability and scalability. It integrates well with .NET 6 and WPF data binding capabilities, allowing for a clear separation between the application's user interface and business logic.

Architecture Type: Windows Presentation Foundation

WPF was chosen because of its extensive interactive UI development features and its smooth interaction with both .NET 6 and the MVVM architecture. The WPF data binding and templating features strengthen and future-proof the user interface. Furthermore, because of its resource management and interoperability with other technologies, it performs better than other frontend technologies like Windows Forms.

## Technical Requirements

- Usage of the latest version of Visual Studio is mandatory for development
- Utilisation of C# is mandatory for all coding activities

- Utilisation of .NET 6 or higher is required for the framework
- Usage of older interface frameworks like WinForms is discouraged (instead use a framework with XAML)
- Usage of local database solutions is mandatory when necessary (use SQLite or SQL Server Express)
- Usage of version control systems is mandatory (use Git or Subversion)
- The final product must be easily set up and run on a standard Windows environment

## Development timeline

**Week 1:** Project setup, requirements gathering, start document

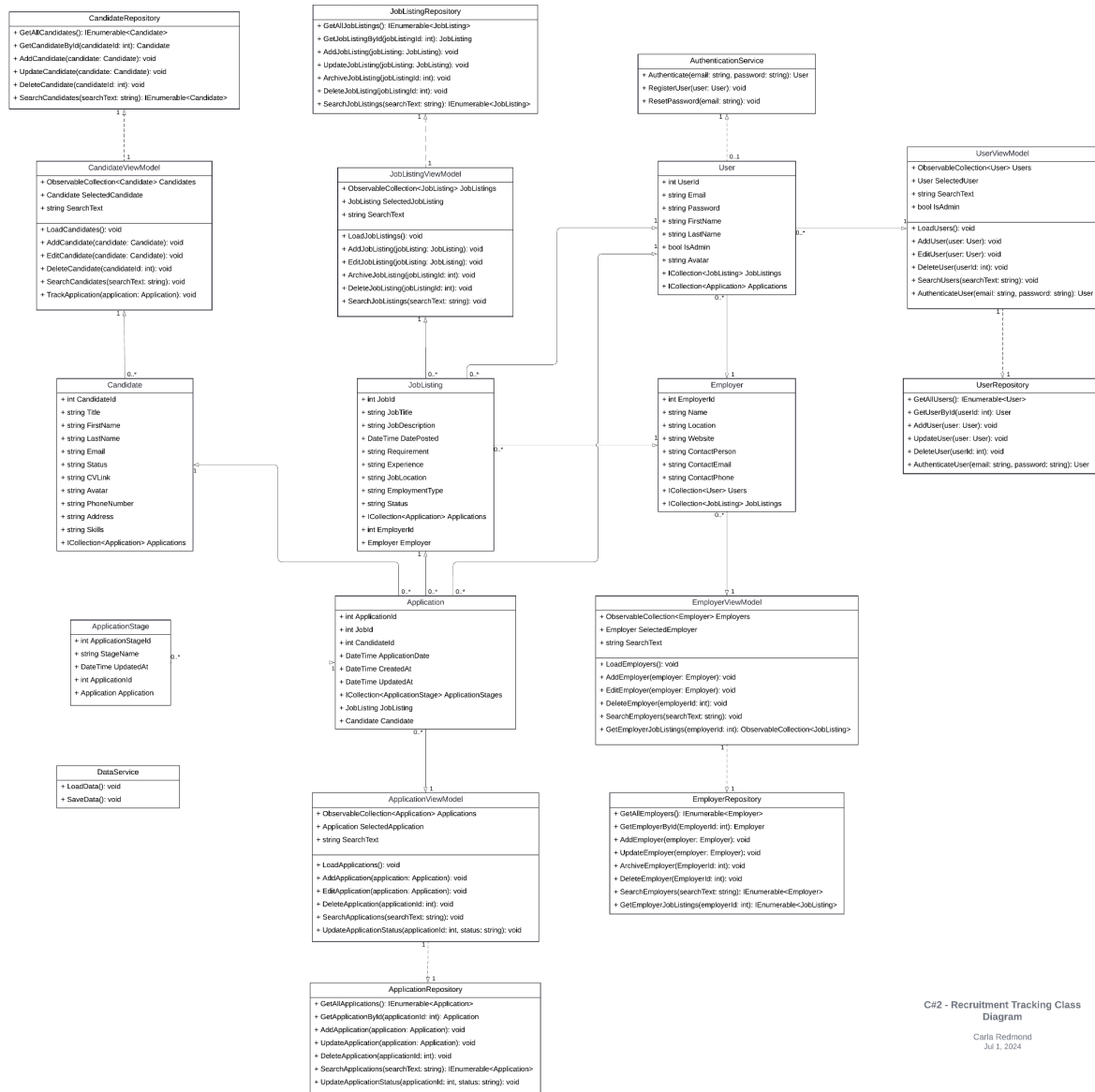
**Week 2:** Database design and backend setup

**Week 3:** Frontend development and basic functionality implementation

**Week 4:** Integration, testing, and bug fixing

**Week 5:** Final presentation preparation and project submission


## Class Diagram



**C#2 - Recruitment Tracking Class  
Diagram**

Carla Redmond  
Jul 1, 2024

Recruiting - Log In






[forgot password?](#)


Log In

Register

Recruiting - Register





Register

Cancel

