Diagram 1: User & Identity Module

This diagram shows the core user management, authentication, and role-based profiles.
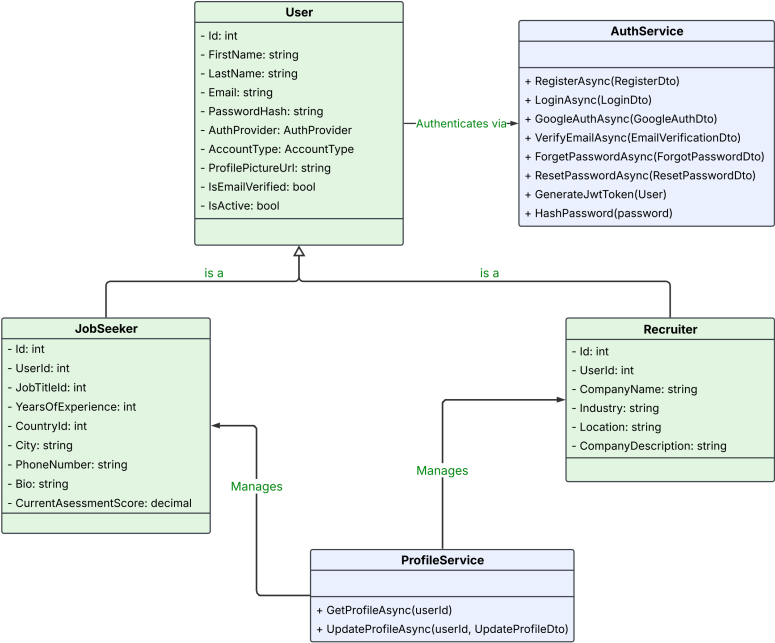
User
- Id: int
- FirstName: string
- LastName: string
- Email: string
- PasswordHash: string
- AuthProvider: AuthProvider
- AccountType: AccountType
- ProfilePictureUrl: string
- IsEmailVerified: bool
- IsActive: bool

—Authenticates via→

AuthService
+ RegisterAsync(RegisterDto)
+ LoginAsync(LoginDto)
+ GoogleAuthAsync(GoogleAuthDto)
+ VerifyEmailAsync(EmailVerificationDto)
+ ForgetPasswordAsync(ForgotPasswordDto)
+ ResetPasswordAsync(ResetPasswordDto)
+ GenerateJwtToken(User)
+ HashPassword(password)

is a / is a

JobSeeker
- Id: int
- UserId: int
- JobTitleId: int
- YearsOfExperience: int
- CountryId: int
- City: string
- PhoneNumber: string
- Bio: string
- CurrentAsessmentScore: decimal

Recruiter
- Id: int
- UserId: int
- CompanyName: string
- Industry: string
- Location: string
- CompanyDescription: string

Manages / Manages

ProfileService
+ GetProfileAsync(userId)
+ UpdateProfileAsync(userId, UpdateProfileDto)

Diagram 2: JobSeeker Profile Module

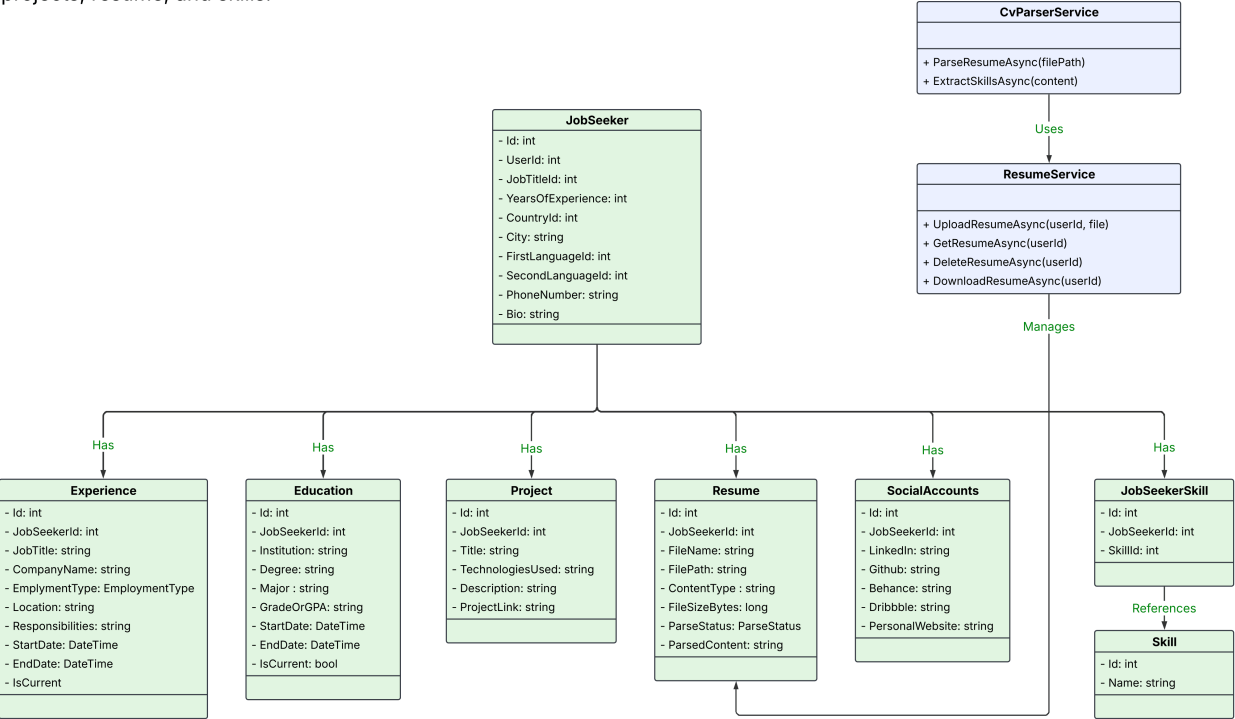This diagram shows the job seeker's portfolio components: experience, education, projects, resume, and skills.

CvParserService
+ ParseResumeAsync(filePath)
+ ExtractSkillsAsync(content)

Uses

ResumeService
+ UploadResumeAsync(userId, file)
+ GetResumeAsync(userId)
+ DeleteResumeAsync(userId)
+ DownloadResumeAsync(userId)

JobSeeker
- Id: int
- UserId: int
- JobTitleId: int
- YearsOfExperience: int
- CountryId: int
- City: string
- FirstLanguageId: int
- SecondLanguageId: int
- PhoneNumber: string
- Bio: string

Manages

Has / Has / Has / Has / Has / Has

Experience
- Id: int
- JobSeekerId: int
- JobTitle: string
- CompanyName: string
- EmploymentType: EmploymentType
- Location: string
- Responsibilities: string
- StartDate: DateTime
- EndDate: DateTime
- IsCurrent

Education
- Id: int
- JobSeekerId: int
- Institution: string
- Degree: string
- Major : string
- GradeOrGPA: string
- StartDate: DateTime
- EndDate: DateTime
- IsCurrent: bool

Project
- Id: int
- JobSeekerId: int
- Title: string
- TechnologiesUsed: string
- Description: string
- ProjectLink: string

Resume
- Id: int
- JobSeekerId: int
- FileName: string
- FilePath: string
- ContentType : string
- FileSizeBytes: long
- ParseStatus: ParseStatus
- ParsedContent: string

SocialAccounts
- Id: int
- JobSeekerId: int
- LinkedIn: string
- Github: string
- Behance: string
- Dribbble: string
- PersonalWebsite: string

JobSeekerSkill
- Id: int
- JobSeekerId: int
- SkillId: int

References

Skill
- Id: int
- Name: string

Diagram 3: Assessment Module

This diagram shows the skill verification system through dynamic quizzes.

AssessmentService

+ StartAssessmentAsync(jobSeekerId)
+ SubmitAnswerAsync(attemptId, dto)
+ CompleteAssessmentAsync(attemptId)
- CalculateScores(attempt)
- SelectQuestions(roleFamily)

JobSeeker
- Id: int
- CurrentAssessmentScore: decimal
- LastAssessmentDate: DateTime

Takes

Manages

Selects Qestions

AssessmentAttempt
- Id: int
- JobSeekerId: int
- JobTitleId: int
- OverallScore: decimal
- TechnicalScore: decimal
- SoftSkillsScore: decimal
- AssessmentStatus: AssessmentStatus
- TimeLimitMinutes: int
- TotalQuestions: int
- QuestionsAnswered: int
- ScoreExpiresAt: DateTime
- IsActive: bool

for Role

JobTitle
- Id: int
- Title: string
- RoleFamily: string

Contains

AssessmentAnswer
- Id: int
- AssessmentAttemptId: int
- QuestionId: int
- SelectedAnswerIndex: int
- IsCorrect: bool

Answers

AssessmentQuestion
- Id: int
- QuestionText: string
- Category: QuestionCategory
- RoleFamily: JobTitleRoleFamily
- SkillId: int
- Difficulty: QuestionDifficulty
- SeniorityLevel: ExperienceSeniorityLevel
- Options: string
- CorrectAnswerIndex: int
- Explanation: string

Tests

Skill
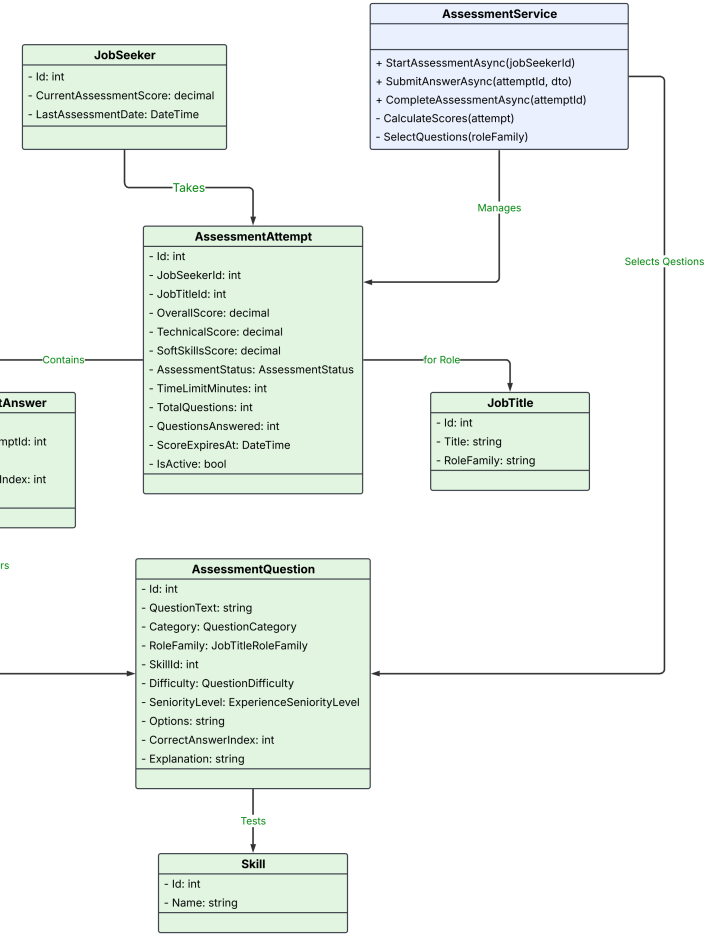- Id: int
- Name: string

Diagram 4: Job Posting & Matching Module

This diagram shows how recruiters post jobs and how the AI-powered matching generates recommendations.

Recruiter
- Id: int
- UserId: int
- CompanyName: string
- Industry: string
- Location: string
- CompanyDescription: string

MatchingService

+ GenerateRecommendationsAsync(jobId)
+ GetJobRecommendationsAsync(jobSeekerId)
+ CalculateMatchScoreAsync(jobSeekerId, jobId)
+ CalculateSkillMatch(jobSeekerSkills, jobSkills)
+ CalculateExperienceScore(yearsOfExp, minRequired)
+ ApplyAssessmentBoost(score, assessment)

JobService

+ CreateJobAsync(recruiterId, CreateJobDto)
+ UpdateJobAsync(recruiterId, jobId, UpdateJobDto)
+ GetJobAsync(jobId)
+ GetRecruiterJobsAsync(recruiterId)
+ DeactivateJobAsync(recruiterId, jobId)

Posts

Manages

Generates

Recommendation
- Id: int
- JobId: int
- JobSeekerId: int
- MatchScore: decimal

Recommends Candidate

Analyzes

Compares

Job
- Id: int
- RecruiterId: int
- Title: string
- Description: string
- Requirements: string
- EmploymentType: EmploymentType
- MinYearsOfExperience: int
- Location: string
- IsActive: bool

for Job

JobSeeker
- Id: int
- YearsOfExperience: int
- CurrentAssessmentScore: decimal

Has

JobSeekerSkill
- Id: int
- JobSeekerId: int
- SkillId: int

Requires

JobSkill
- Id: int
- JobId: int
- SkillId: int

References

References

Skill
- Id: int
- Name: string