Software Design Document for Connekt

6.1 Introduction

This document presents the architecture and detailed design for the software of Connekt, a web application designed to support new graduates and job seekers in navigating their career search while identifying ideal locations to live. Connekt integrates job search functionality using the Indeed API, a personalized dashboard for job tracking, and Esri's web mapping tool to visualize job locations relative to essential lifestyle services.

6.1.1 System Objectives

The objective of Connekt is to create an intuitive platform that helps users:

- Search for jobs based on skills and preferences using the Indeed API.
- Identify ideal living locations based on job proximity and lifestyle needs.
- Utilize Esri's web mapping tool to visualize essential services like grocery stores, entertainment, and public transport options.
- Create a dashboard of the surrounding neighborhood using Esri's Business Analyst API
- Use ChatGPT API for suggestions to initialize the search.

6.1.2 Hardware, Software, and Human Interfaces

- **Hardware Interfaces:** Connekt is a web-based platform accessible via desktops, laptops, etc.. potential for mobile app support
- **Software Interfaces:**
    - **Indeed API** – Retrieves job listings based on user preferences.
    - **Esri Web Mapping API** – Provides location-based insights for job locations and essential services.
    - **Database (Firebase)** – Stores user profiles and application tracking data.
    - **Frontend (React.js/Vite)** – Provides an interactive and user-friendly experience.
- **Human Interfaces:**
    - Interactive map interface for users to explore job locations and nearby amenities.

6.3 CSC and CSU Descriptions

This section provides the detailed design of Connekt, outlining the Computer Software Components (CSC) and their respective Computer Software Units (CSU). Connekt follows a **Model-View-Controller (MVC) architecture**, breaking down into the following CSCs:

1. **Frontend CSC (Connekt UI Layer)**
    - Handles user interactions and data visualization.
    - Uses React.js for rendering pages dynamically.
    - Includes user dashboards, job search views, and map visualizations.
2. **Backend CSC (Connekt API Layer)**
    - Processes job search requests and user data.
    - Handles authentication, job tracking, and database interactions.
3. **Database CSC (Connekt Data Storage)**
    - Stores user profiles, saved jobs, and preference settings.
    - Uses Firebase for real-time updates.
4. **Integration CSC (External API Interfaces)**
    - Connects with external services such as the Indeed API for job listings.
    - Uses Esri's web mapping API for location-based services.
    - Implements secure API request handling.

Each of these CSCs is composed of multiple CSUs (classes and modules), which will be detailed in the following sections.

6.3.1 Detailed Class Descriptions

The following sections provide the details of all classes used in the Connekt application.

6.3.1.1 User Class

- **Description:** Represents a user of the Connekt application, storing personal details and job preferences.
- **Fields:**
    - `name`: User's full name.
    - `email`: Email address for authentication.
    - `preferences`: User-defined job search and location preferences.
    - `savedJobs`: Array of job postings saved by the user.?
- **Methods:**
    - `saveJob(jobId)`: Adds a job to the saved list.
    - `updatePreferences(prefs)`: Updates job search and location preferences.

6.3.1.2 Job Class

- **Description:** Represents a job listing retrieved from the Indeed API.
- **Fields:**

- `title`: Job title.
- `company`: Name of the hiring company.
- `location`: Geographic location of the job.
- `salary`: Estimated salary (if available).

6.3.1.3 Dashboard Class

- **Description:** Manages the user's job search dashboard, displaying saved jobs, applications, and recommendations.
- **Fields:**
  - `userId`: ID of the associated user.
  - `savedJobs`: Jobs saved by the user.
  - `recommendedJobs`: AI-suggested jobs based on preferences.
- **Methods:**
  - `getSavedJobs()`: Returns the list of saved jobs.
  - `getRecommendedJobs()`: Fetches job recommendations.

6.3.2 Detailed Interface Descriptions

This section describes how the different components of Connekt communicate with each other.

- **Frontend ↔ Backend Interface**
  - The frontend interacts with the backend via REST API endpoints.
  - Requests include user authentication, job searches, and data retrieval.
- **Backend ↔ External APIs**
  - Indeed API fetches job listings based on query parameters.
  - Esri API provides map-based job location visualization.
  - Secure authentication is implemented for API calls.

6.3.3 Detailed Data Structure Descriptions

This section provides details on the data structures used in Connekt, including data storage formats, API responses, and other structured data representations.

6.3.3.1 User Data Structure

- **Purpose:** Stores user information, preferences, and saved job listings.

Structure:
json

```
{
  "user_id": "UUID",
  "name": "string",
  "email": "string",
  "preferences": {
    "job_title": "string",
    "location": "string",
  },
}
```

- **Usage:** Used for authentication, job filtering, and dashboard display.

6.3.3.2 Job Data Structure

- **Purpose:** Represents job listings fetched from the **Indeed API** and stored for user reference.
- **Format:** JSON object.

Structure:
json

```
{
  "job_id": "string",
  "title": "string",
  "location": "string",
  "salary": "string",?

}
```

- **Usage:** Stored temporarily in the backend and displayed in job search results.

6.3.3.3 Map Data Structure

- **Purpose:** Stores geospatial data for job locations, lifestyle services, and user-selected regions.
- **Format:** GeoJSON for Esri API integration.

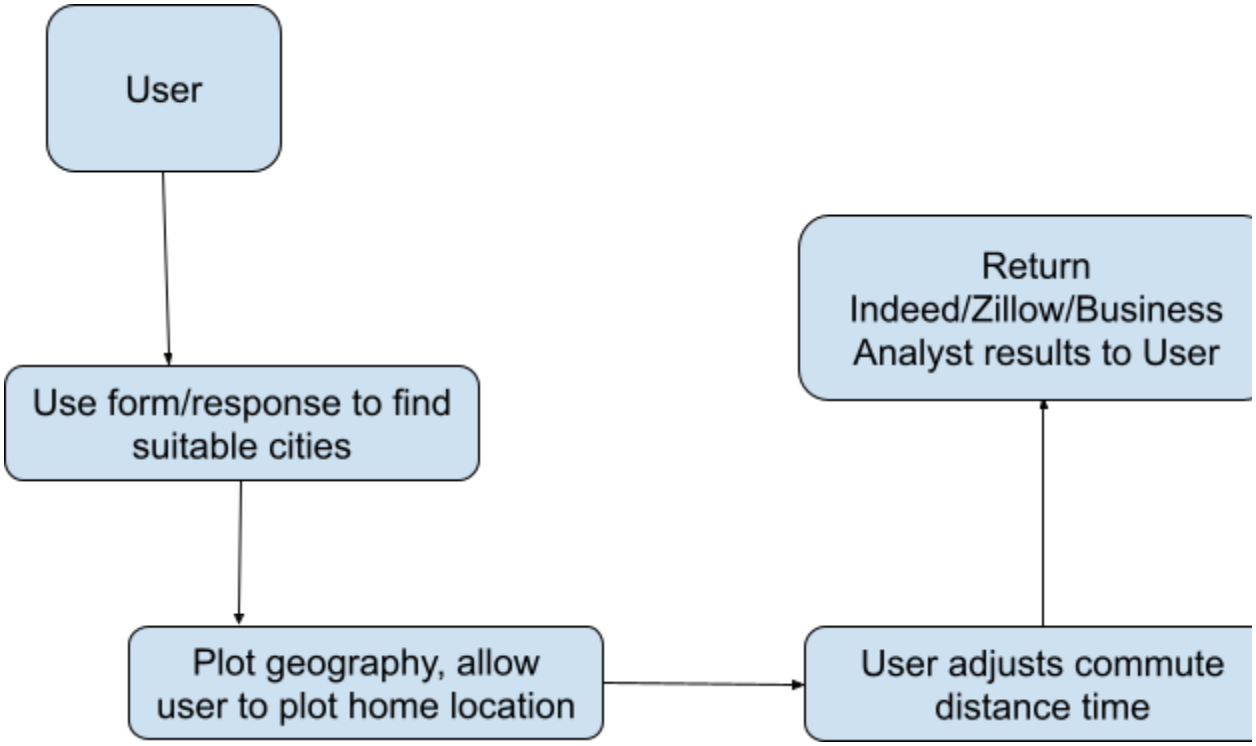
Structure:
json

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [longitude, latitude]
      },
      "properties": {
        "name": "string",
        "category": "string"
      }
    }
  ]
}
```

-
- **Usage:** Used for map visualization, filtering job locations, and lifestyle features.

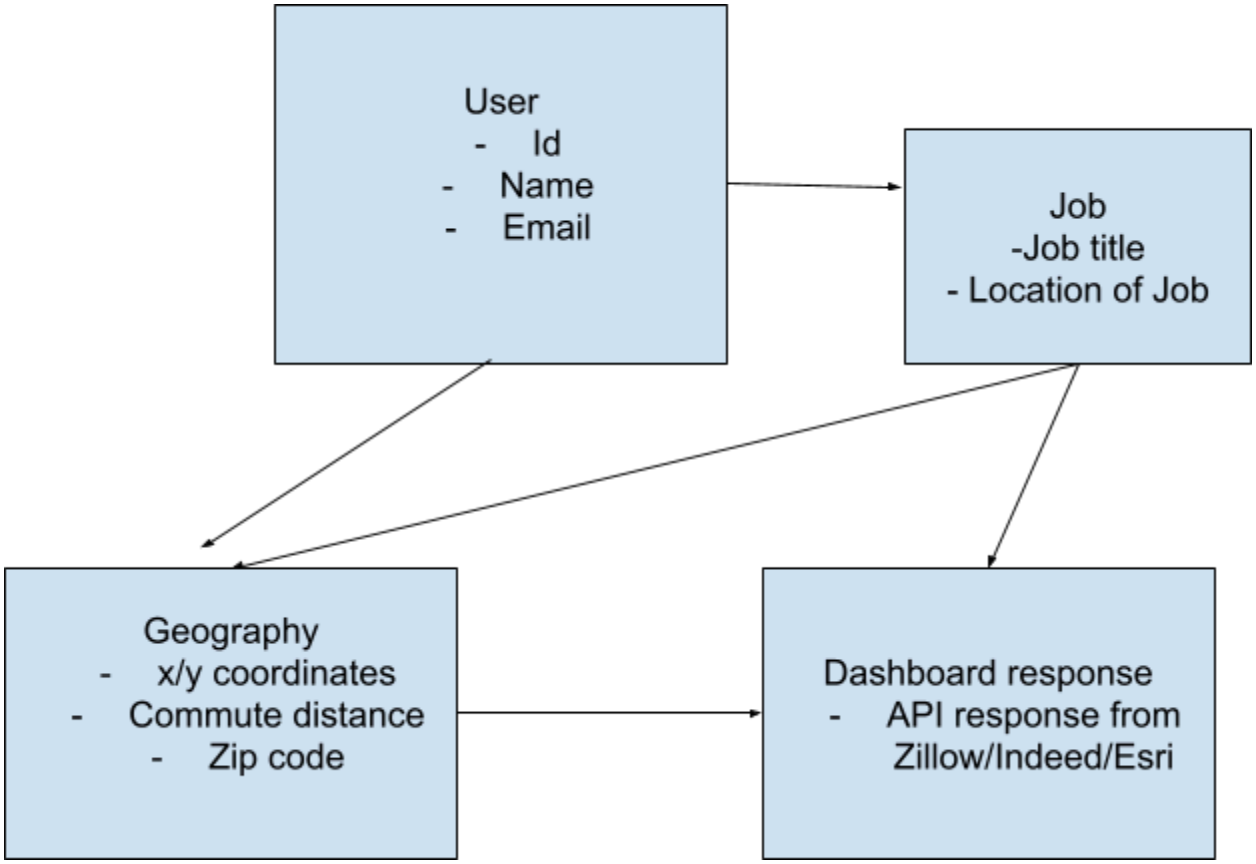
6.3.4 Detailed Design Diagrams

6.3.4.1 Use Case Diagram



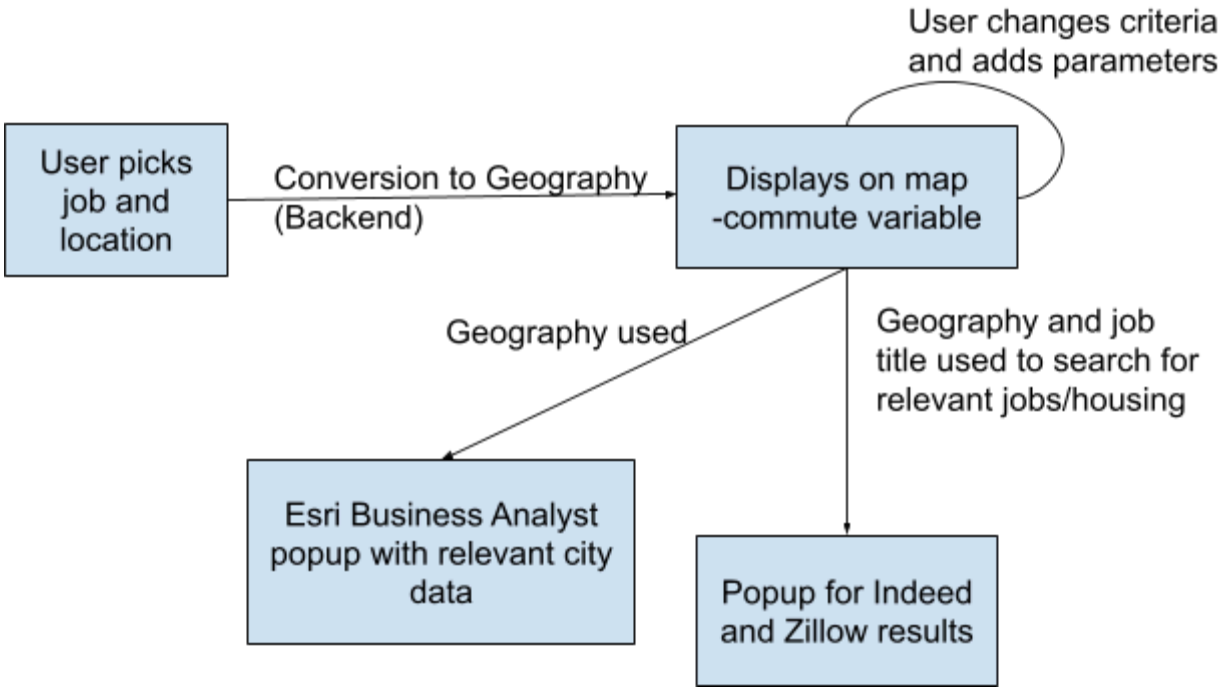
- How users interact with the application

6.3.4.2 Class Diagram

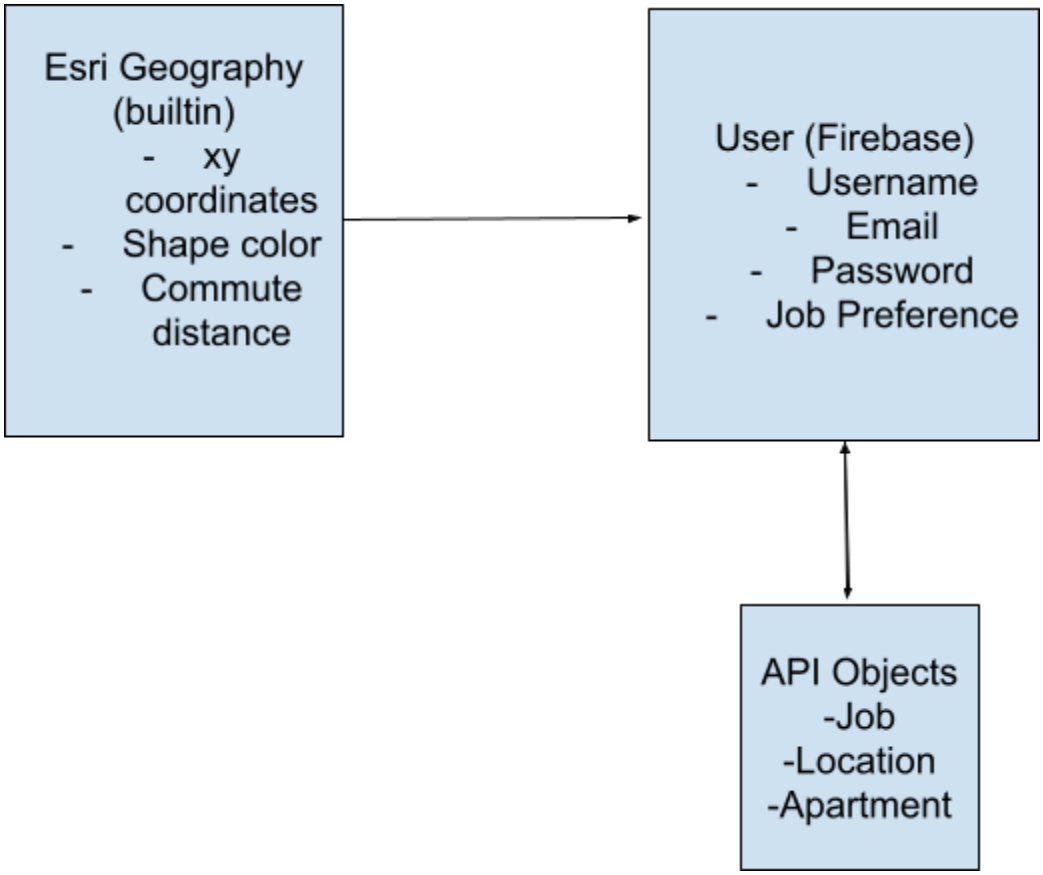


- The relationships between key classes, such as User, Job, Dashboard, and API handlers.

6.3.4.3 Sequence Diagram



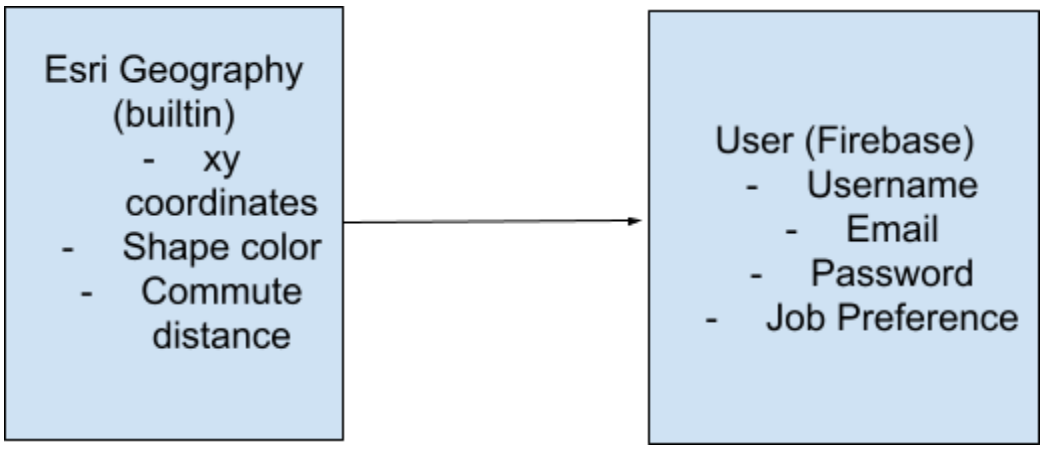
6.3.4.4 ER Diagram (For Database Design)



6.4 Database Design and Description

Since Connekt uses a database to store user data and saved jobs, this section details its structure and access methods.

6.4.1 Entity-Relationship Diagram (ERD)



6.4.2 Database Access

- **Technology:** Firebase for relational storage
- **Access Method:**
    - The backend API interacts with Firebase

6.4.3 Database Security

- **Authentication:** Users authenticate via Firebase Auth
- **Rate Limiting:** API requests are rate-limited.
