VU

Virtual University

CARE SEEKERS
GROUP ID: S2302148F7

DESIGN DOCUMENT

PREPARED BY

ALI BILAL AHMED
BC180404460

SUPERVISE BY

FAIZAN TAHIR
VIRTUAL UNIVERSITY

CARE SEEKERS

Revision History

Date (dd/mm/yyyy)	Version	Description	Author
15/07/2023	1.0	The design document serves as a comprehensive guide for the development of the Care Seekers project, providing detailed information about its various components, functionalities, and design aspects. The design document includes the following sections Introduction, Entity Relationship Diagram (ERD), Sequence Diagrams, Architecture Design Diagram, Class Diagram, Database Design, Interface Design and Test Cases. The design document serves as a reference throughout the development process, helping the development team understand the system requirements, make informed decisions, and ensure the successful implementation of the Care Seekers project.	Ali Bilal BC180404460

CARE SEEKERS

Care Seekers Design Document Contents

CARE SEEKERS

1.Introduction of Design Document

Introduction

Care Seeker design document is a comprehensive and structured document that outlines the design aspects of a Care Seeker development project. It serves as a roadmap and reference for the development team, supervisor, and virtual university of Pakistan, providing a detailed overview of the system's architecture, database, functionality, Interface and behavior. The design document acts as a bridge between the Care Seeker project requirements and the actual implementation, ensuring a clear and consistent understanding of the Care Seeker project's design.

The Care Seeker design document captures the design decisions and considerations made during the design phase of the Care Seeker web application development lifecycle. It provides a detailed description of how the system will be structured, how its components will interact, and how the desired functionality will be achieved. The document serves as a guide for the development team, helping them understand the system's requirements and implement it according to the agreed-upon design principles.

Inclusions in the Design Document

The Care Seeker design document includes various components and sections that cover different aspects of the Care Seeker web application development project. These typically include:

- **Entity Relationship Diagram (ERD):** A visual representation of the relationships between entities in the Care Seeker web application, helping to understand the data model and associations between different entities.
- **Sequence Diagrams:** Illustrations of the interactions and task sequences between Care Seeker web application components, showcasing the flow of events for each use case.
- **Architecture Design Diagram:** An overview of the Care Seeker web application's architecture, including the separation of presentation layer, application layer, and data layer.

- **Class Diagram:** A representation of the structure and relationships between classes in the Care Seeker web application, providing a high-level view of the Care Seeker web application's object-oriented design.
- **Database Design:** The structure of the database and relationships between tables, outlining how data is stored and retrieved within the Care Seeker web application.
- **Interface Design:** Screenshots or mockups of the graphical user interface (GUI) of the Care Seeker web application, showcasing the visual representation and layout of key features.
- **Test Cases:** Detailed scenarios and steps to validate the Care Seeker web application 's functionality, ensuring that it meets the specified requirements.

Benefits of a Design Document

The design phase of a Care Seeker web application development project offers several benefits, including the following Benefits.

- **Clear Understanding:** The design document ensures that myself, supervisor and Virtual University of Pakistan have a clear understanding of the Care Seeker web application's design, database, functionality, interface and behavior. It serves as a common reference point and helps align expectations.

- **Planning and Documentation:** The Care Seeker design document provides a structured approach to planning and documenting the Care Seeker web application's design decisions, architecture, and interfaces. It helps in organizing the development effort and provides clarity to the myself and supervisor.

- **Early Issue Identification:** By examining the Care Seeker design early in the development process, potential issues, challenges, and risks can be identified. This allows for proactive mitigation and problem-solving before significant resources are invested in implementation.

- **Effective Communication:** The Care Seeker design document serves as a communication tool, facilitating effective collaboration and communication among

supervisor, Virtual University of Pakistan, and any other stakeholders. It helps in conveying design concepts and decisions clearly and consistently.

- **Informed Decision Making:** The design phase allows for the evaluation of different design alternatives and trade-offs. It enables informed decision-making based on factors such as feasibility, scalability, maintainability, and user experience.

- **Flexibility and Adaptability:** The Care Seeker design document provides a foundation for iterative refinement and adjustments. It allows for flexibility and adaptability to accommodate changing requirements and feedback throughout the development process.

- **Quality Assurance:** A well-designed Care Seeker web application is more likely to be reliable, maintainable, and scalable. The design phase allows for careful consideration of quality attributes and helps lay the groundwork for a high-quality end product.

Purpose of a Design Document

The purpose of the design phase in a Care Seeker web application development project is to plan and document the Care Seeker web application's design, ensuring a clear understanding of the project's requirements, objectives, and constraints. It serves as a crucial intermediary step between requirements analysis and system implementation. The main objectives of the design phase include:

Translating Requirements into a Concrete Design: The design phase bridges the gap between high-level requirements and the actual system design. It transforms abstract requirements into concrete design decisions, enabling their implementation.

Enabling Efficient Development: The Care Seeker design document acts as a guide for the myself, providing me with a well-defined structure and design decisions. It helps streamline the development process and ensures consistency across the team.

Reducing Development Risks: By carefully considering Care Seeker design aspects, potential risks and challenges can be identified early on. This allows for proactive mitigation and reduces the chances of costly rework or late-stage design changes.

Facilitating Collaboration and Communication: The Care Seeker design document serves as a reference for all stakeholders, promoting effective collaboration and communication among stakeholders, supervisor and Virtual University of Pakistan. It helps ensure a shared understanding of the Care Seeker web application's design and functionality.

Providing Documentation for Future Reference: The Care Seeker design document serves as a valuable source of documentation for future maintenance, enhancements, or re-evaluation of the Care Seeker web application. It provides a record of the design decisions and justifications made during the development process.

2. Entity Relationship Diagram (ERD)

Conceptual Entity-Relationship Diagram

Conceptual Entity-Relationship Diagram (ERD) of Care seeker
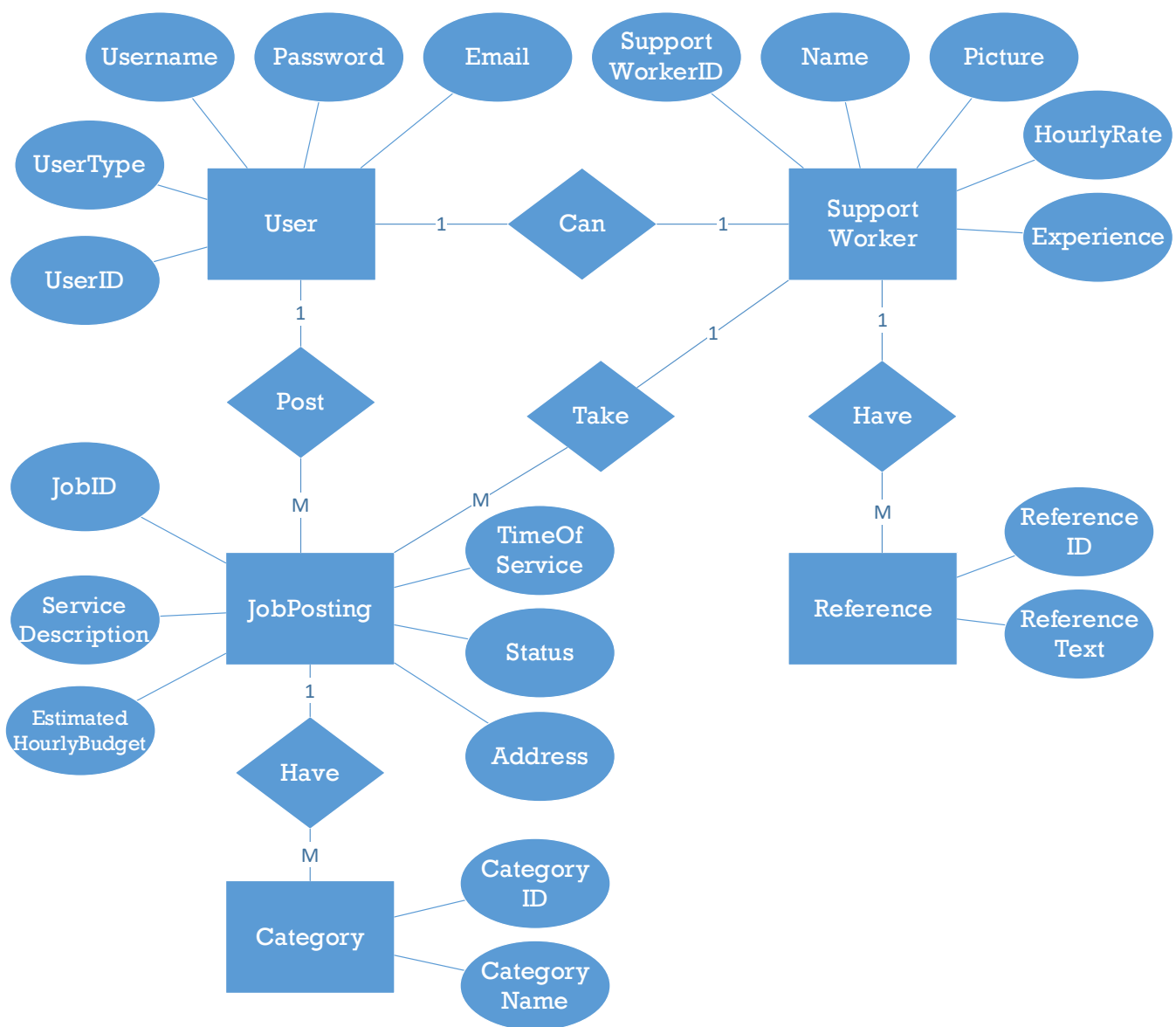
Explanation:
Entities:

1. **User:**
   - UserID: A unique identifier for each user in the system.
   - Username: The username chosen by the user for login and identification.
   - Password: The user's password for authentication.
   - Email: The email address associated with the user's account for communication and account recovery.
   - UserType: Indicates whether the user is a Support Worker or a Care Seeker. This attribute helps distinguish between the two types of users.

2. **SupportWorker:**
   - SupportWorkerID (Primary Key): A unique identifier for each support worker in the system.
   - Name: The name of the support worker.
   - Picture (or ImageURL): A picture or URL to an image of the support worker.
   - HourlyRate: The rate at which the support worker charges for their services on an hourly basis.
   - Experience: Information about the support worker's relevant experience.

3. **Reference:**
   - ReferenceID : A unique identifier for each reference or recommendation.
   - ReferenceText: A text field where references or recommendations for a support worker can be provided.

4. **Category:**
   - CategoryID (Primary Key): A unique identifier for each category of services.
   - CategoryName: The name of the service category, such as Baby Care, Cooking, etc. This helps categorize and group services.

5. **JobPosting:**
   - JobID: A unique identifier for each job posting.
   - ServiceDescription: A detailed description of the service or job being posted by a care seeker.
   - Address: The location or address where the service is required.

- EstimatedHourlyBudget: The budget the care seeker has allocated for the service, typically on an hourly basis.
- TimeOfService: The date and time when the service is needed.
- Status: Indicates the status of the job posting (e.g., Open or Accepted).

Relationships:

1. **User has a one-to-one relationship with SupportWorker:**
   - This relationship indicates that each user in the system, whether they are a care seeker or a support worker, can have a one-to-one association with a support worker profile. In other words, every user has the option to become a support worker, and this connection allows a user to access their support worker profile if they choose to be one.

2. **SupportWorker has a one-to-many relationship with Reference:**
   - This relationship signifies that a support worker can have multiple references or recommendations associated with their profile. These references can come from different sources or individuals who have worked with the support worker in the past. It allows potential clients to see the support worker's track record.

3. **CareSeeker (UserID) has a one-to-many relationship with JobPosting:**
   - This relationship shows that a care seeker can post multiple job listings or service requests. Each job posting is associated with a particular care seeker, allowing them to manage multiple requests for different services.

4. **Category has a one-to-many relationship with JobPosting:**
   - This relationship represents the categorization of job postings. Each job posting falls into a specific service category (e.g., Baby Care, Cooking), and multiple job postings can belong to the same category. It helps organize and filter service requests.

CARE SEEKERS

Physical Entity-Relationship Diagram

Physical Entity-Relationship Diagram (ERD) of Care seeker
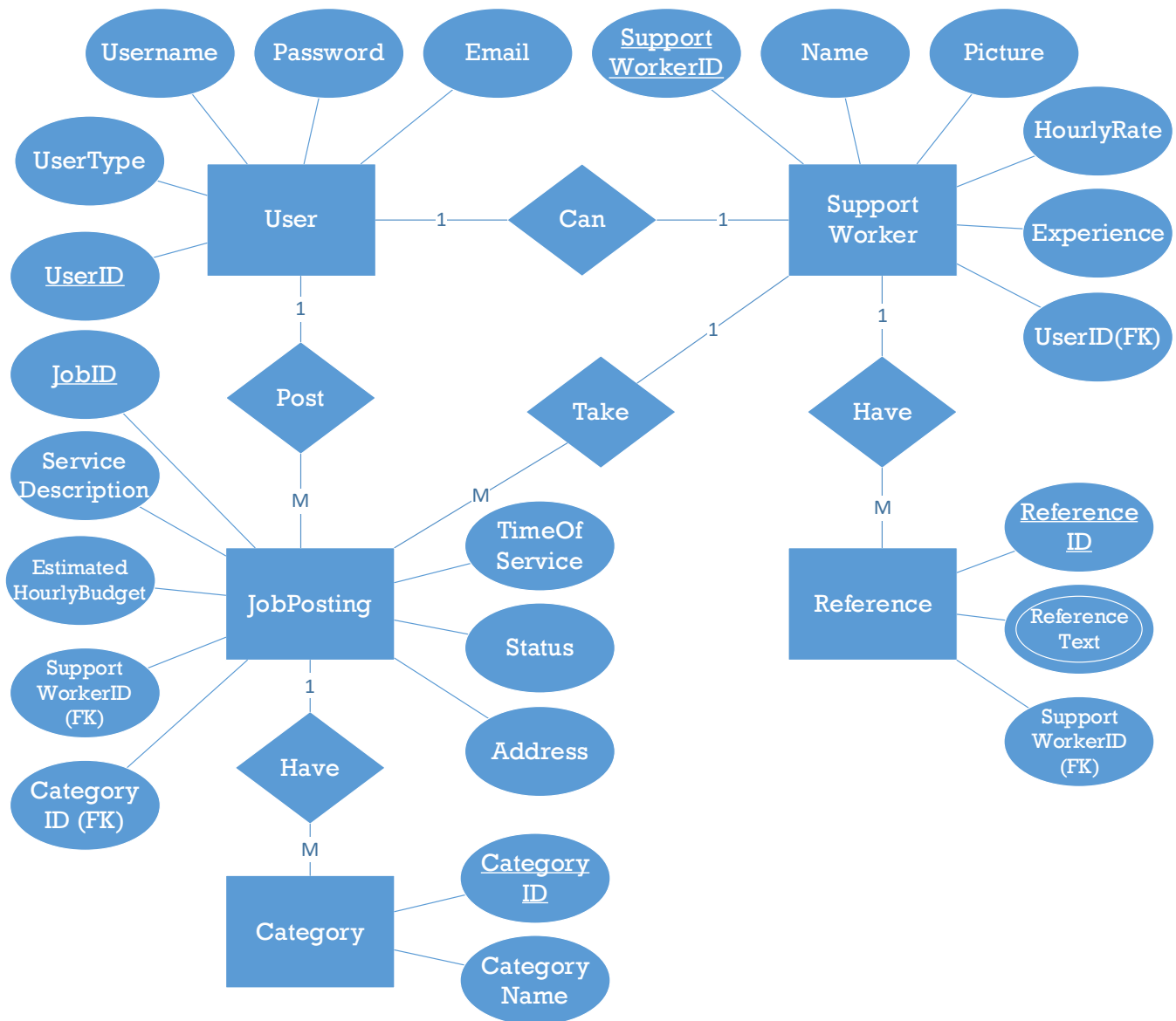
Explanation:

Entities and Attributes:

1. **User:**
   - UserID (Primary Key, INT): A unique identifier for each user in the system.
   - Username (VARCHAR): The username chosen by the user for login and identification.
   - Password (VARCHAR): The user's password for authentication.
   - Email (VARCHAR): The email address associated with the user's account for communication and account recovery.
   - UserType (ENUM: Care Seeker, Support Worker): Indicates whether the user is a Care Seeker or a Support Worker. This attribute helps distinguish between the two types of users.

2. **SupportWorker:**
   - SupportWorkerID (Primary Key, INT): A unique identifier for each support worker's profile.
   - WorkerName (VARCHAR): The name of the support worker.
   - Picture (BLOB or URL): An image representing the support worker.
   - HourlyRate (DECIMAL): The rate at which the support worker charges for their services on an hourly basis.
   - Experience (TEXT): Information about the support worker's relevant experience.
   - UserID (Foreign Key to User): Links the support worker's profile to their user account, allowing them to log in and manage their profile.

3. **Reference:**
   - ReferenceID (Primary Key, INT): A unique identifier for each reference or recommendation.
   - ReferenceText (TEXT): A text field where references or recommendations for a support worker can be provided.
   - SupportWorkerProfileID (Foreign Key to SupportWorkerProfile): Links the reference to the specific support worker it pertains to. Multiple references can be associated with a single support worker.

4. **Category:**
   - CategoryID (Primary Key, INT): A unique identifier for each category of services.
   - CategoryName (VARCHAR): The name of the service category, such as "Baby Care" or "Cooking." This helps categorize and group services.

5. **JobPosting:**
   - JobID (Primary Key, INT): A unique identifier for each job posting.
   - ServiceDescription (TEXT): A detailed description of the service or job being posted by a care seeker.
   - Address (VARCHAR): The location or address where the service is required.
   - EstimatedHourlyBudget (DECIMAL): The budget allocated by the care seeker for the service, typically on an hourly basis.
   - TimeOfService (DATETIME): The date and time when the service is needed.
   - Status (ENUM: Open, Accepted): Indicates the status of the job posting, whether it's open for applications or has been accepted by a support worker.
   - CareSeekerID (Foreign Key to User): Links the job posting to the care seeker who posted it.
   - CategoryID (Foreign Key to CareCategory): Links the job posting to the category of services it belongs to.

Relationships:

1. **User has a one-to-one relationship with SupportWorker:**
   - This relationship indicates that each user in the system, whether they are a care seeker or a support worker, can have a one-to-one association with a support worker profile. In other words, every user has the option to become a support worker, and this connection allows a user to access their support worker profile if they choose to be one.

2. **SupportWorker has a one-to-many relationship with Reference:**
   - This relationship signifies that a support worker can have multiple references or recommendations associated with their profile. These references can come from different sources or individuals who have worked with the support worker in the past. It allows potential clients to see the support worker's track record.

3. **CareSeeker (UserID) has a one-to-many relationship with JobPosting:**
   - This relationship shows that a care seeker can post multiple job listings or service requests. Each job posting is associated with a particular care seeker, allowing them to manage multiple requests for different services.

4. **Category has a one-to-many relationship with JobPosting:**
   - This relationship represents the categorization of job postings. Each job posting falls into a specific service category (e.g., Baby Care, Cooking), and multiple job postings can belong to the same category. It helps organize and filter service requests.

CARE SEEKERS

3. Sequence Diagrams

Sign up

Sequence Diagrams of Sign Up
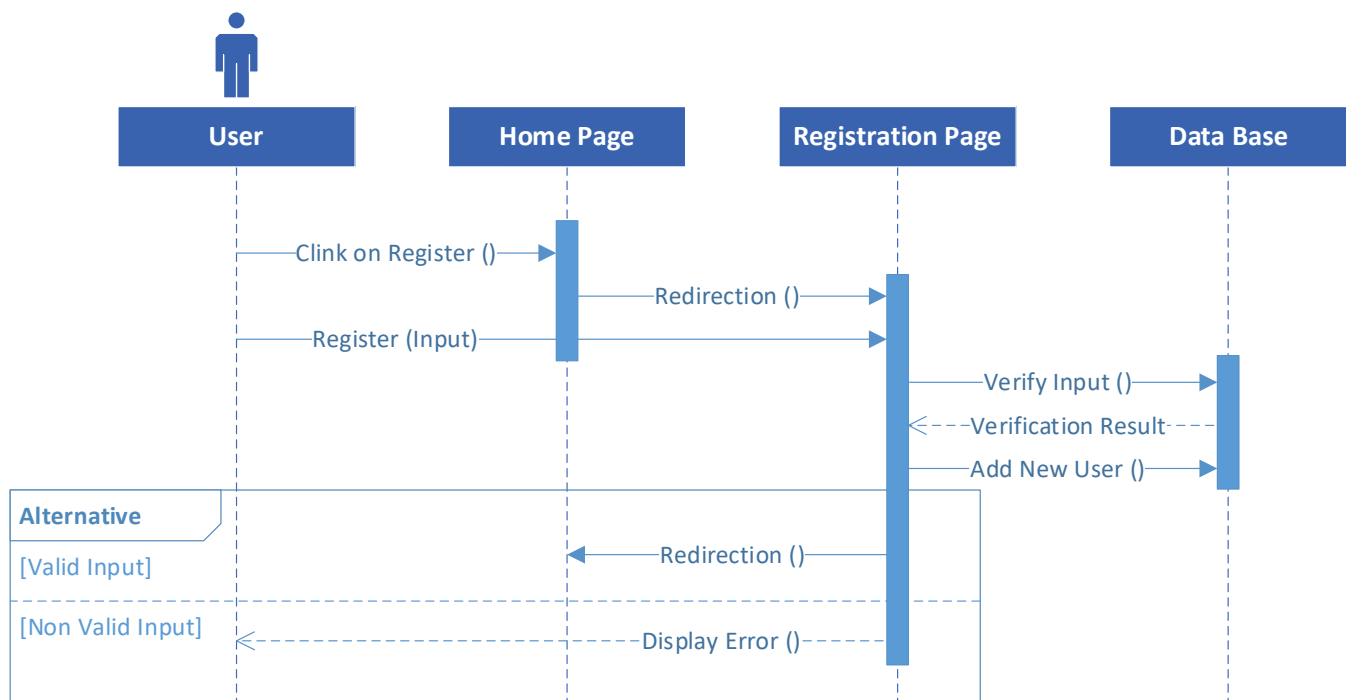
In this sequence diagram:

- The actor (User) initiates the sign-up process by clicking on register button on the Care Seekers website Homepage.
- The actor (User) redirected to registration page.
- The Care Seekers website displays the sign-up form for the user to enter their details.
- The user enters their sign-up details, including their name, email, password for care seeker and more detail for Support worker like biodata, including personal information, relevant qualifications, upload a profile picture, relevant experience, hourly rate for the services and at least two references for verification purposes.
- The Care Seekers website validates the sign-up details entered by the user.
- The Care Seekers website generates a unique User ID for the new user.

CARE SEEKERS

- The Care Seekers website creates a user account using the provided details and generated User ID.
- The Care Seekers will redirect to Homepage website where it displays a sign-up success message to the user, indicating that their account has been created successfully as support worker or care seeker.
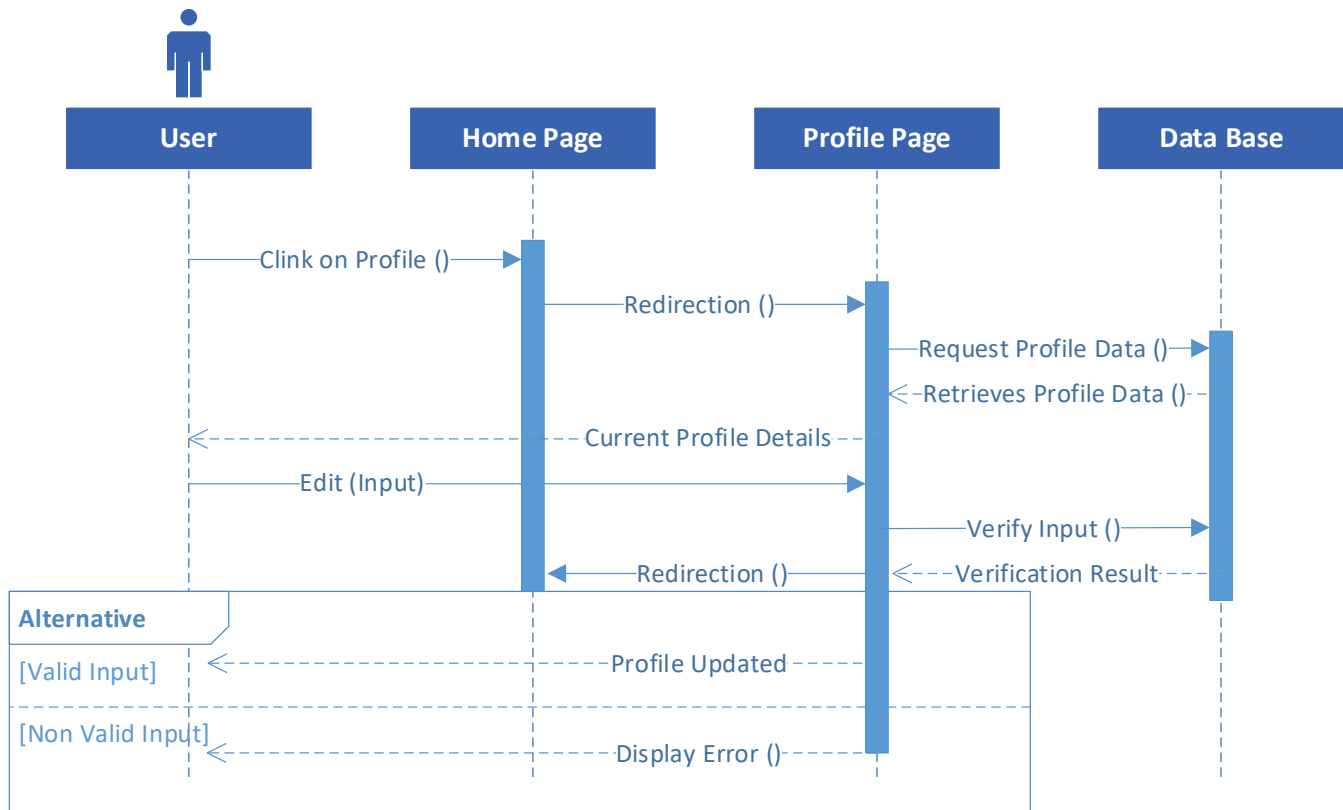
Login

Sequence Diagrams of Log In
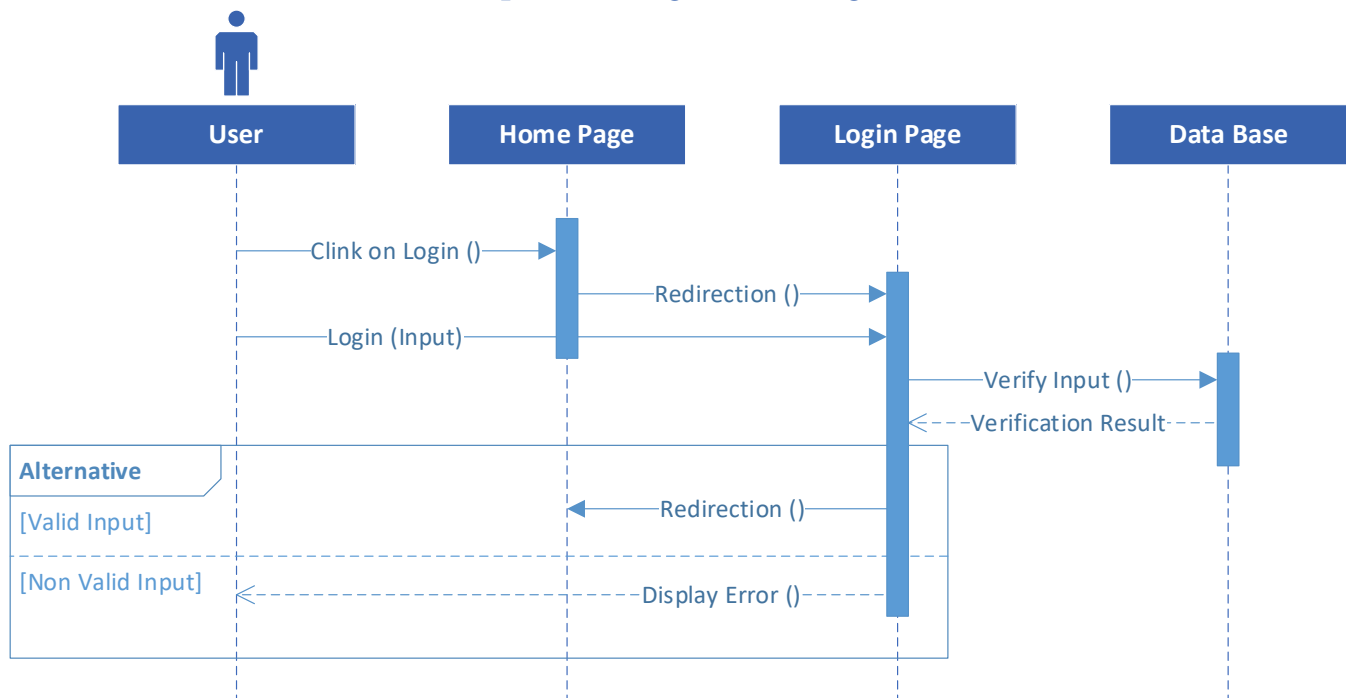
In this sequence diagram:

- The actor (Support Worker User / Care Seeker User) initiates the login process by clinking on login button on home page of the Care Seekers website.
- The Care Seekers website displays the login page with login form for the user to enter their login credentials.
- The user enters their login credentials, including their email and password.
- The Care Seekers website validates the login credentials entered by the user.
- The Care Seekers website authenticates the user by verifying their credentials against the stored user data.
- If the login credentials are valid, the Care Seekers website grants access to the user.
- If the login credentials are invalid, the Care Seekers website will display login error.

Manage Profile

Sequence Diagrams of Manage Profile

In this sequence diagram:

- The actor (Support Worker User / Care Seeker User) initiates the "Manage Profile" use case by clicking Profile button on the Care Seekers website.
- The Care Seekers website displays Profile page with the user's profile, showing their current profile details.
- The user views their profile details on the Care Seekers website.
- The Care Seekers website retrieves the user's profile data from the storage or database.
- The Care Seekers website displays the profile data to the user.
- The user requests to edit their profile on the Care Seekers website.
- The Care Seekers website validates the profile updates entered by the user.
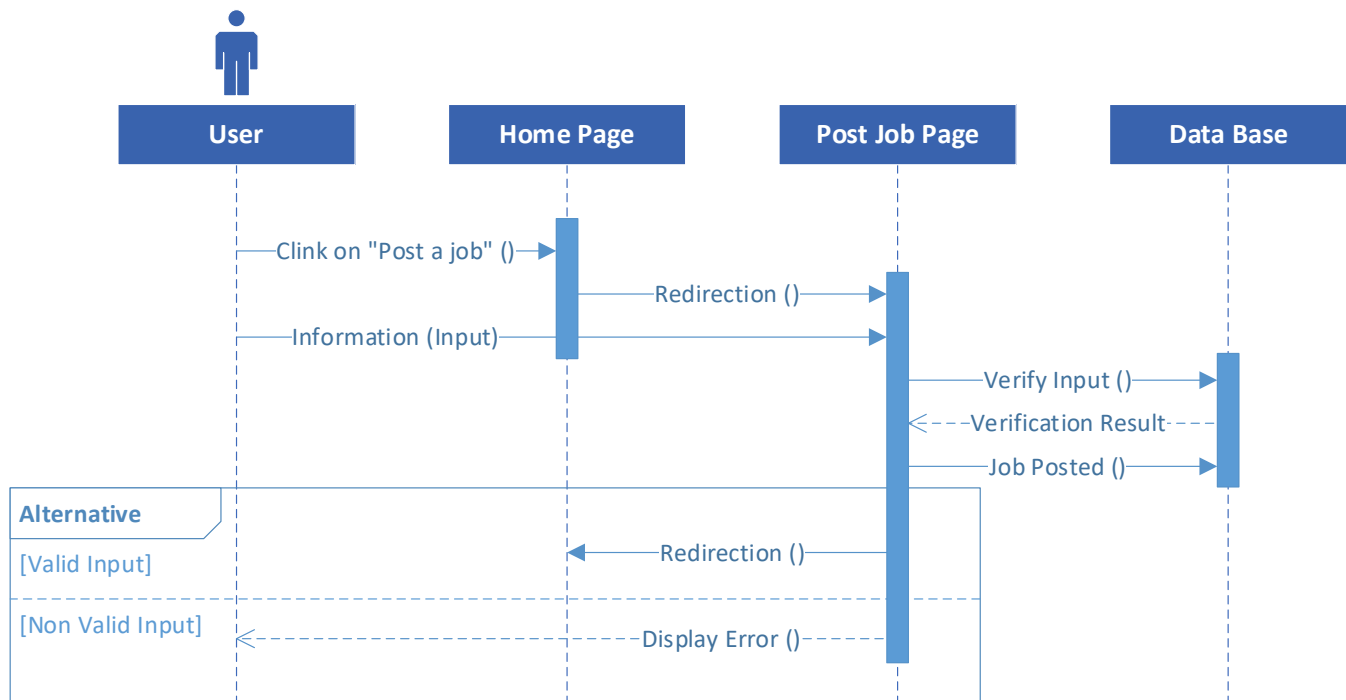- The Care Seekers website updates the user's profile data with the validated updates.

- The Care Seekers website displays a success message to the user, confirming that their profile has been updated successfully.

Posting Job

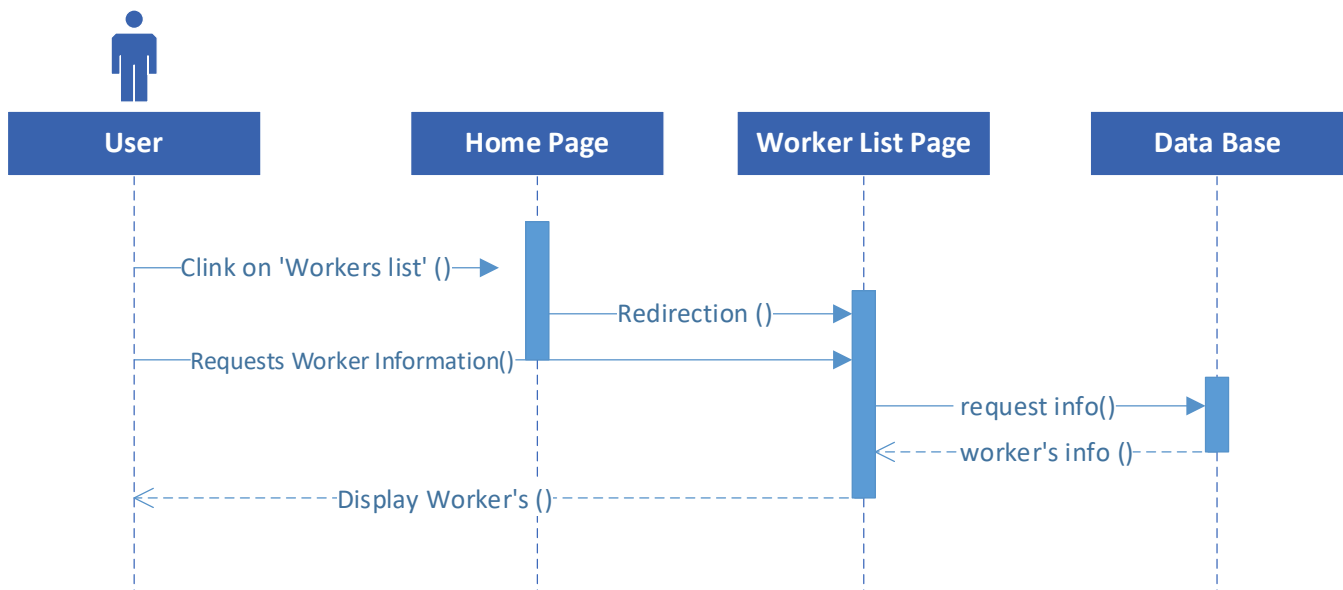Sequence Diagrams of Posting Job

In this sequence diagram:

- The actor Care Seeker User initiates the Posting Job by clinking on 'Post a job' button on home page of the Care Seekers website.
- The Care Seekers website displays the 'Posting Job' page with job form for the user to enter their job Information.
- The user enters their job Information, including Service Description, Address, Estimated Hourly Budget, Time of Service, and Category.
- The Care Seekers website validates the information entered by the user.
- If the information are valid, the Care Seekers website will post a job.
- If the information are invalid, the Care Seekers website will display error.

Display Worker

Sequence Diagrams of Display Worker

In this sequence diagram:

- The actor Care Seeker User initiates the Display worker by clinking on 'worker's list' button on home page of the Care Seekers website
- Care Seeker initiates a request to Display Worker Information by interacting with the Worker's list page.
- The Database receives the request and prepares to retrieve worker information.
- The worker's information is sent back to the Worker's list page.
- The Worker's list page receives the worker's information and prepares to display it.
- The worker's information is displayed to the Care Seeker.

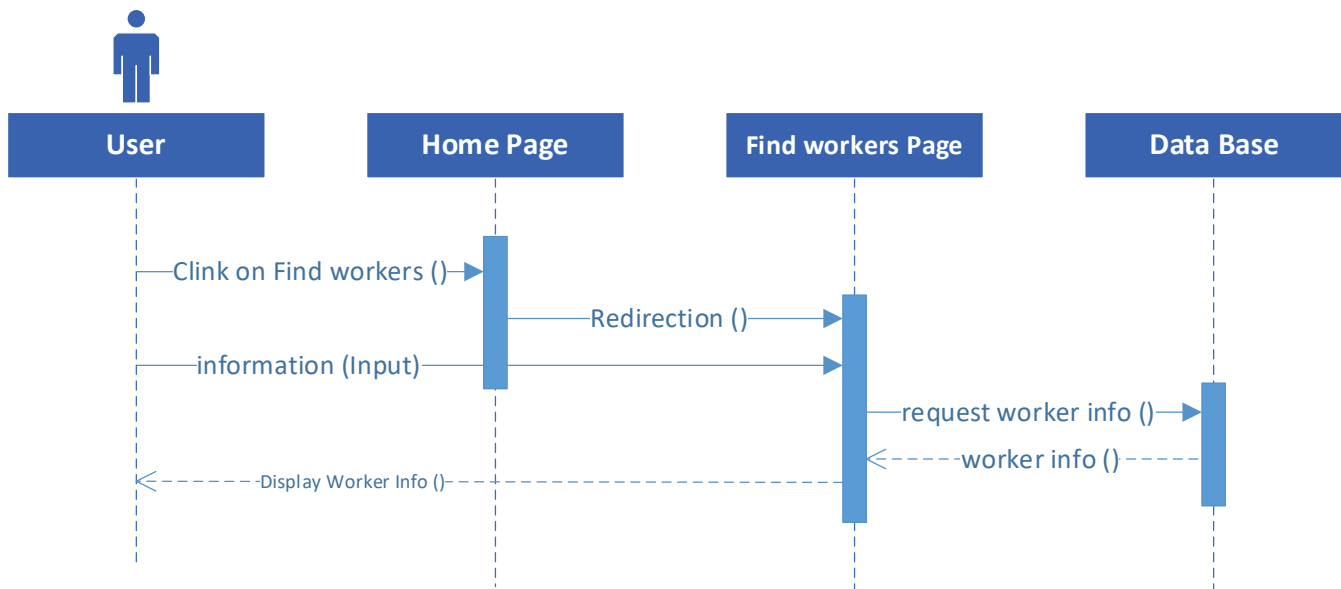**Search Worker**

Sequence Diagrams of Search Worker

- In this sequence diagram:
  The actor Care Seeker User initiates the Search Worker by clinking on 'Find Worker' button on home page of the Care Seekers website
- The Care Seekers website displays the 'Find Worker' page with Search box for the user to enter their Information.
- The Care Seeker website sends a Search Worker Info request to the Database.
- The Database receives the request and prepares to retrieve worker information.
- The worker's information is sent back to the Find Worker page page.
- The Find Worker pag receives the worker's information and prepares to display it.
- The worker's information is displayed to the Care Seeker.

4. Architecture Design Diagram

Diagram

Care Seeker Three-Tier Architecture

User

Presentation Layer
Web Client || Mobile Client || Desktop Client

Application Layer
API Gateway | | Services | | Logic Business
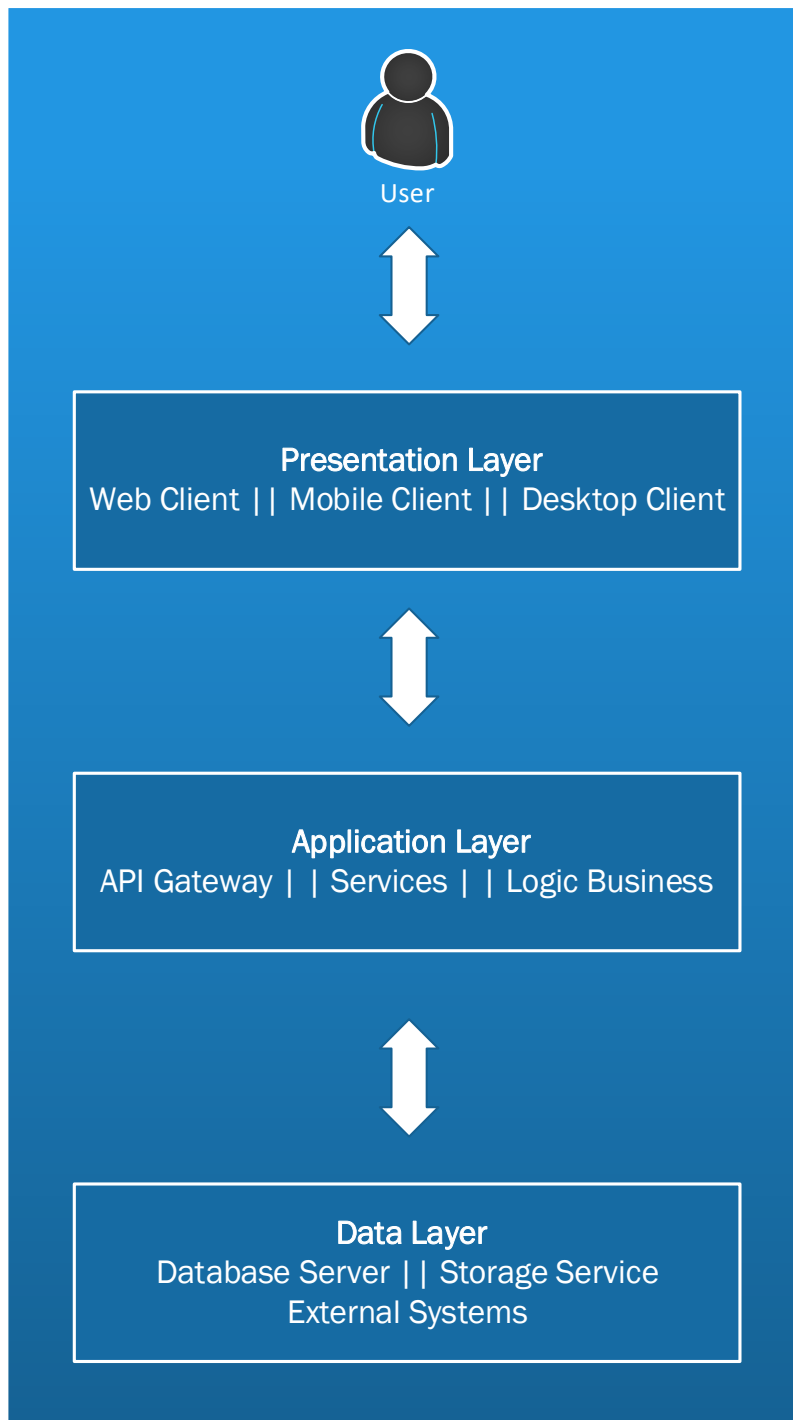
Data Layer
Database Server || Storage Service
External Systems

Explanation:

The architecture design depicted in the diagram is commonly known as a "Three-Tier Architecture" or "Three-Layer Architecture." It is a widely used architectural pattern in software development that separates the components of a system into three distinct layers: presentation, application, and data.

The three-tier architecture promotes modularity, scalability, and maintainability by separating the concerns of the system into separate layers, each with its own responsibilities. This design pattern allows for flexibility and easier management of the system's components, as changes or updates in one layer typically do not affect the other layers.

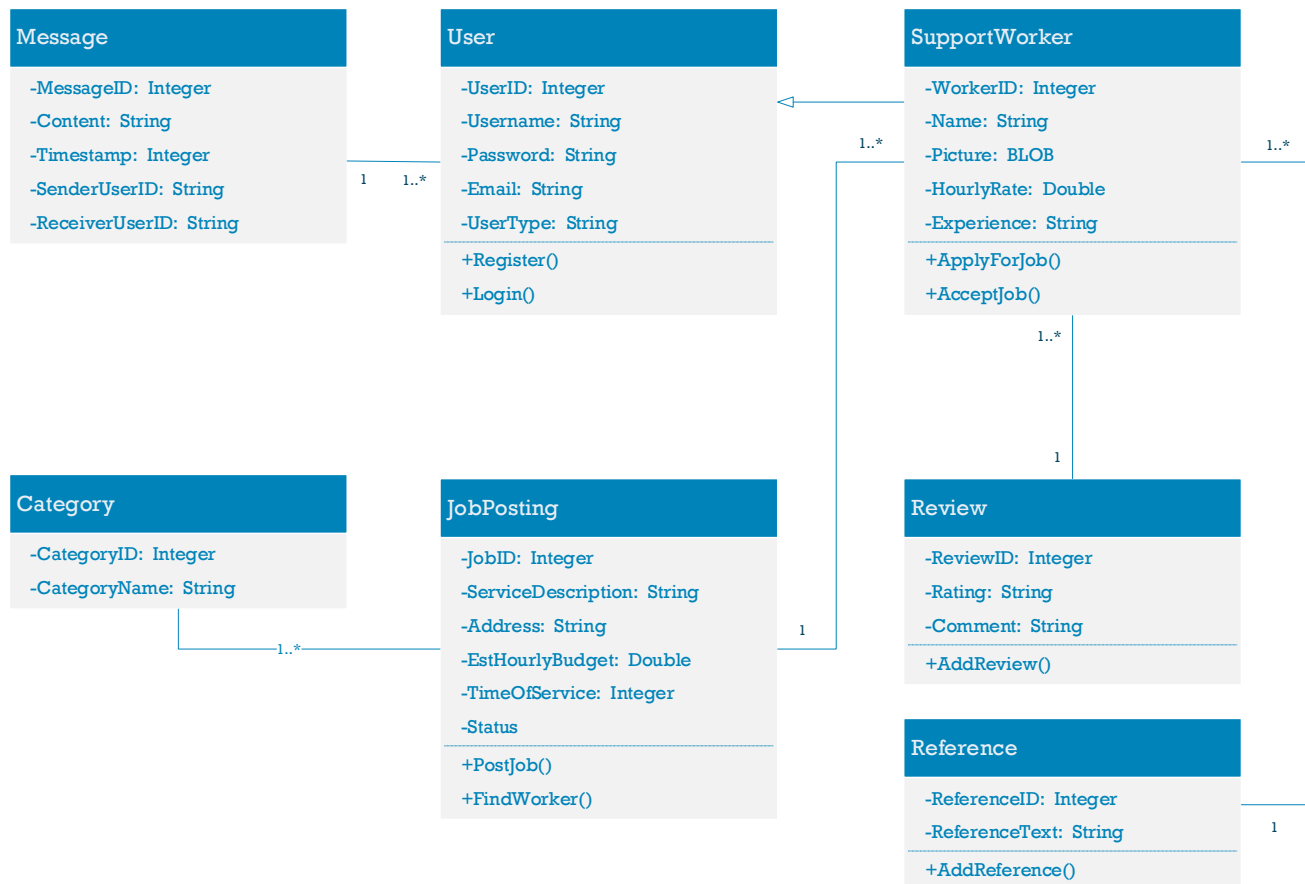In this tiered architecture design diagram:

1. Presentation Layer: This layer represents the user interface (UI) components of the system. It consists of various client applications such as web, mobile, and desktop clients, which interact with the users and provide the interface for accessing the system's functionalities.

2. Application Layer: This layer contains the business logic and services that process the user requests received from the presentation layer. It consists of components such as API gateways, services, and business logic modules. The API gateway acts as an entry point, routing and handling the incoming requests from different clients.

3. Data Layer: This layer manages the storage and retrieval of data used by the system. It includes components such as databases (e.g., relational or NoSQL databases), storage services (e.g., for file storage), and external systems (e.g., APIs or integrations with external services).

The tiered architecture design separates the concerns of the system into distinct layers, promoting modularity, scalability, and maintainability. The presentation layer handles user interaction, the application layer handles business logic and processing, and the data layer manages data storage and retrieval.

5. Class Diagram

Diagram

Class Diagram of Project Care Seeker

Message

-MessageID: Integer
-Content: String
-Timestamp: Integer
-SenderUserID: String
-ReceiverUserID: String

User

-UserID: Integer
-Username: String
-Password: String
-Email: String
-UserType: String

+Register()
+Login()

SupportWorker

-WorkerID: Integer
-Name: String
-Picture: BLOB
-HourlyRate: Double
-Experience: String

+ApplyForJob()
+AcceptJob()

Category

-CategoryID: Integer
-CategoryName: String

JobPosting

-JobID: Integer
-ServiceDescription: String
-Address: String
-EstHourlyBudget: Double
-TimeOfService: Integer
-Status

+PostJob()
+FindWorker()

Review

-ReviewID: Integer
-Rating: String
-Comment: String

+AddReview()

Reference

-ReferenceID: Integer
-ReferenceText: String

+AddReference()

1   1..*   1..*   1..*   1..*   1   1   1..*   1

Class Explanation:

1. **User:**
   - Represents users of the system, including Care Seekers and Support Workers.
   - Attributes: UserID, Username, Password, Email, UserType.
   - Methods: Register() - for user registration, Login() - for user login.

2. **SupportWorker:**
   - Represents support workers who provide services.
   - Attributes: WorkerID, Name, Picture, HourlyRate, Experience.
   - Methods: ApplyForJob() - for applying to job postings, AcceptJob() - for accepting job offers.

3. **Category:**
   - Represents service categories, e.g., Baby Care, Cooking.
   - Attributes: CategoryID, CategoryName.

4. **JobPosting:**
   - Represents job postings created by Care Seekers.
   - Attributes: JobID, ServiceDescription, Address, EstHourlyBudget, TimeOfService, Status.
   - Methods: PostJob() - for posting job listings, FindWorker() - for finding support workers.

5. **Review:**
   - Represents reviews and ratings for Support Workers.
   - Attributes: ReviewID, Rating, Comment.
   - Methods: AddReview() - for adding reviews.

6. **Reference:**

- Represents references or recommendations for Support Workers.
- Attributes: ReferenceID, ReferenceText.
- Methods: AddReference() - for adding references.

7. Message:
- Represents messages exchanged between users.
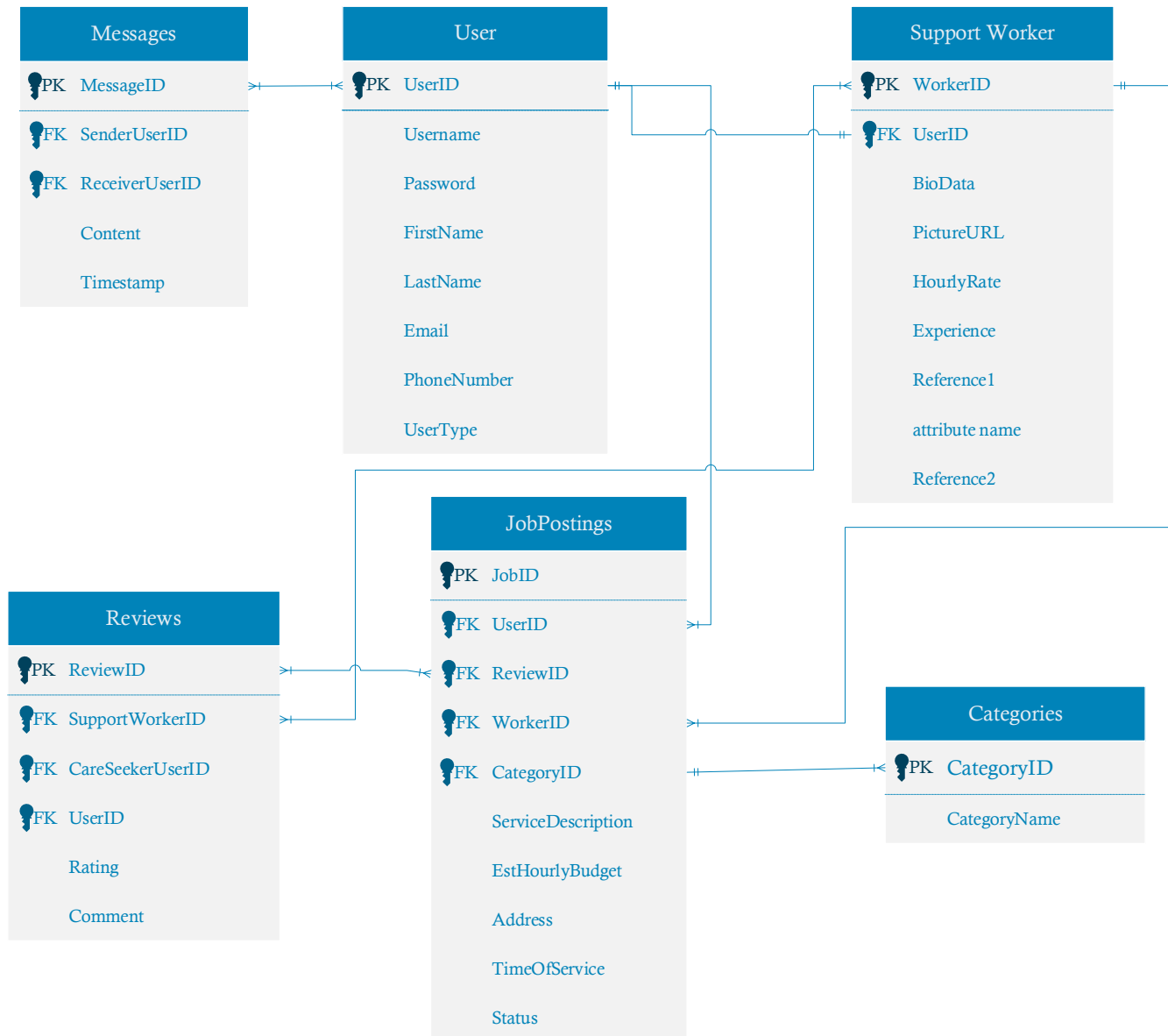- Attributes: MessageID, Content, Timestamp, SenderUserID, ReceiverUserID.

Relationships:

- User and SupportWorker have an inheritance relationship, where SupportWorker is a subclass or derived class of User.
- SupportWorker has a one-to-many relationship with Reference (a Support Worker can have multiple references).
- User has a one-to-many relationship with JobPosting (a Care Seeker can post multiple job listings).
- Category has a one-to-many relationship with JobPosting (job postings are categorized).
- SupportWorker has a one-to-many relationship with Review (a Support Worker can have multiple Review).
- Review has a one-to-one relationship with SupportWorker (each Support Worker can have one review).
- Reference has a one-to-one relationship with SupportWorker (each Support Worker can have one reference).
- User has a many-to-many relationship with Message (users exchange messages).

6. Database Design

Diagram

Database Design Diagram

Messages	
PK	MessageID
FK	SenderUserID
FK	ReceiverUserID
	Content
	Timestamp

User	
PK	UserID
	Username
	Password
	FirstName
	LastName
	Email
	PhoneNumber
	UserType

Support Worker	
PK	WorkerID
FK	UserID
	BioData
	PictureURL
	HourlyRate
	Experience
	Reference1
	attribute name
	Reference2

Reviews	
PK	ReviewID
FK	SupportWorkerID
FK	CareSeekerUserID
FK	UserID
	Rating
	Comment

JobPostings	
PK	JobID
FK	UserID
FK	ReviewID
FK	WorkerID
FK	CategoryID
	ServiceDescription
	EstHourlyBudget
	Address
	TimeOfService
	Status

Categories	
PK	CategoryID
	CategoryName

Explanation:

1. **Users Table:**

Description:
This table stores information about users, including both Care Seekers and Support Workers.

Attributes:

- UserID (Primary Key): Unique identifier for each user.
- Username: User's username for login.
- Password: User's hashed or encrypted password.
- Email: User's email address.
- UserType: Indicates whether the user is a Care Seeker or Support Worker.

2. **SupportWorkers Table:**

Description:
This table stores additional information about Support Workers.

Attributes:

- WorkerID (Primary Key): Unique identifier for each support worker.
- Name: Name of the support worker.
- Picture: Path or URL to the worker's profile picture.
- HourlyRate: Hourly rate charged by the support worker.
- Experience: Information about the worker's experience.

3. **CareCategories Table:**

Description:
This table defines various service categories, such as Baby Care, Cooking, etc.

Attributes:

- CategoryID (Primary Key): Unique identifier for each service category.
- CategoryName: Name of the service category.

4. **JobPostings Table:**

Description:
This table stores information about job postings created by Care Seekers.

Attributes:

- JobID (Primary Key): Unique identifier for each job posting.
- ServiceDescription: Description of the service required.
- Address: Address where the service is needed.
- EstHourlyBudget: Estimated hourly budget for the job.
- TimeOfService: Date and time when the service is required.
- Status: Status of the job posting (e.g., Open, Closed).
- CareSeekerUserID (Foreign Key): Links to the Users table to associate the job posting with a Care Seeker.
- CategoryID (Foreign Key): Links to the CareCategories table to specify the service category.

5. **Reviews Table:**

Description:
This table stores reviews and ratings provided for Support Workers.

Attributes:

- ReviewID (Primary Key): Unique identifier for each review.
- Rating: Numeric rating (e.g., 1 to 5) given by the Care Seeker.
- Comment: Textual comment or review.
- SupportWorkerID (Foreign Key): Links to the SupportWorkers table to associate the review with a specific support worker.
- CareSeekerUserID (Foreign Key): Links to the Users table to identify the Care Seeker who provided the review.

6. **Messages Table:**

Description:
This table stores messages exchanged between users.

Attributes:

- MessageID (Primary Key): Unique identifier for each message.
- Content: Content of the message.
- Timestamp: Date and time when the message was sent.
- SenderUserID (Foreign Key): Links to the Users table to identify the sender of the message.
- ReceiverUserID (Foreign Key): Links to the Users table to identify the receiver of the message.

7. Interface Design

In this section, we present the graphical user interfaces (GUI) of the Care Seeker web application. The following screenshots provide a visual representation of the most important features of the application as envisioned during the design phase. Please note that these designs may be subject to updates during the actual development process.

1. Home Page
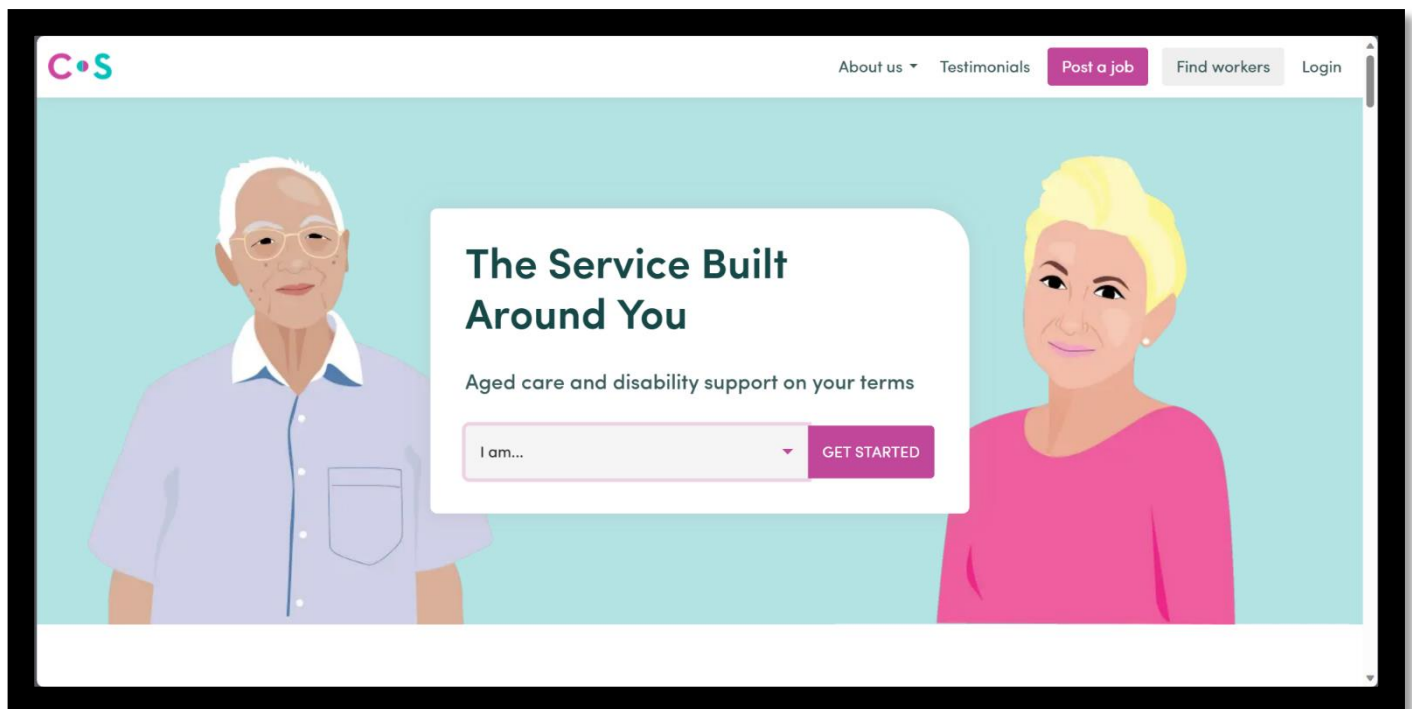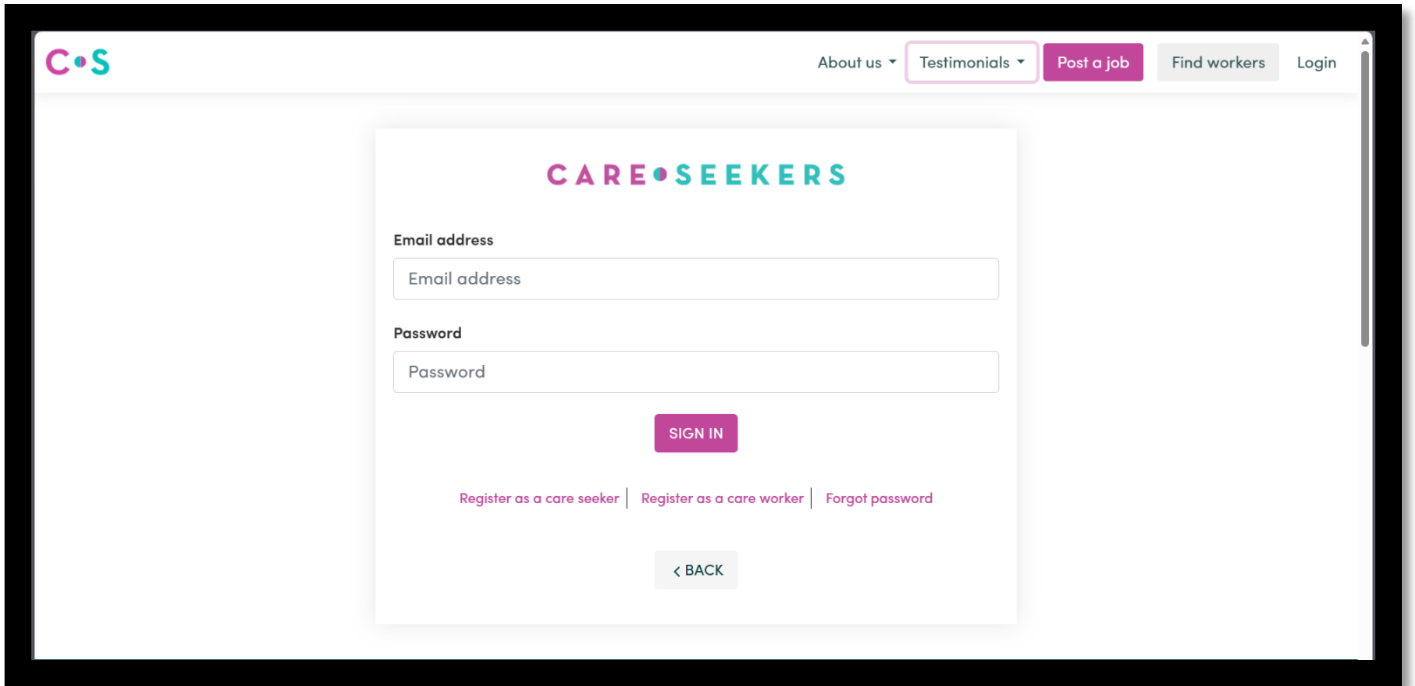
Description:
The Home Page welcomes users to the Care Seeker platform. It features a clean and user-friendly layout with easy navigation options. At the top, you'll find the project logo and a header menu for quick access to key sections of the application, such as posting a job, finding a support worker, reading testimonials, and learning more about the platform.

2. Sign In Page

Description:
The Sign In Page allows registered users to access their accounts securely. Users can enter their usernames and passwords to log in. A "Forgot Password" link is provided for password recovery. Successful sign-in redirects users to their respective dashboards.

3. Find Worker Page

Description:
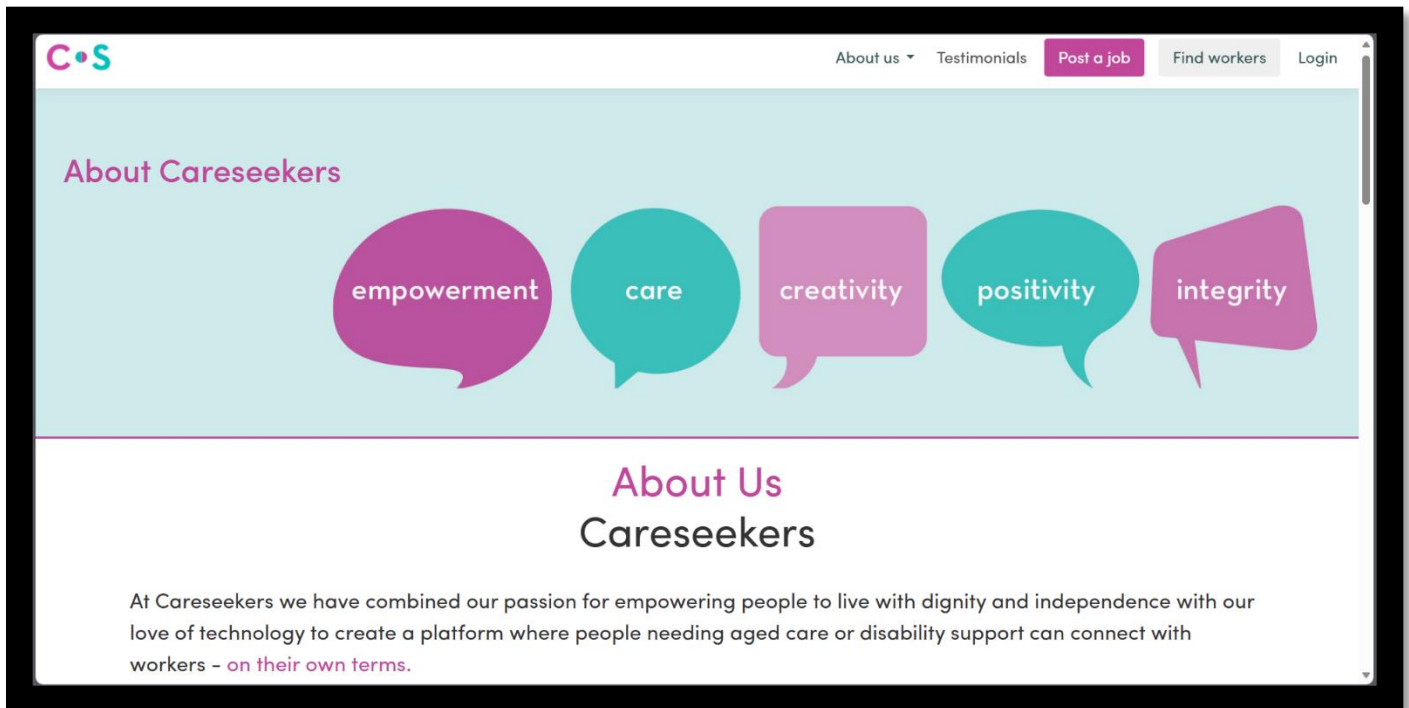The Find Worker Page provides care seekers with the ability to search for support workers based on their specific needs. Users can filter support workers by service category, location, experience, and hourly rate. The search results display support workers who match the selected criteria, making it easy for care seekers to find the right fit.

4. About Us Page

Description:
The About Us Page provides information about the Care Seeker platform. Users can learn about the project's mission, goals, and the team behind its development. This page offers insights into the platform's purpose and the dedication of the team to provide quality caregiving services.

8. Test Cases

Sign Up

Test Case Title	Sign up
Test Case ID	TC-02
Actor(s)	Anonymous user
Pre-Conditions	Anonymous user must have an internet connection.
Task Sequence	
- The user clicked the site link and opened the main page through typing the URL of this web application. - The user clicked on "Register" button for the purpose of registration and then user provided the information required in text fields and check boxes. - The application validated the user information and then registered the user with given information.	
Expected Result	User will be able to register himself.
Final Result	User has registered successfully as Care Seeker/Support Worker.
Tested By	BC180404460

Log In

Test Case Title	Log In
Test Case ID	TC-02
Actor(s)	Care Seeker / Support Worker
Pre-Conditions	Care Seeker or Support Worker user must have an internet connection.
Task Sequence	
- Care Seeker or Support Worker opened the web application, provide login credentials and clicked on login button. - Care Seeker or Support Worker information was validated and he had protected area access while unauthorized access was denied.	
Expected Result	Care Seeker or Support Worker will be able to login himself.
Final Result	Care Seeker or Support Worker has logged in successfully.
Tested By	BC180404460

Manage Profile

Test Case Title	Manage Profile
Test Case ID	TC-03
Actor(s)	Care Seeker or Support Worker
Pre-Conditions	Care Seeker or Support Worker user must have an internet connection. Care Seeker or Support Worker must be logged in.
Task Sequence	
- Care Seeker or Support Worker logged in to the web application through valid login credentials. - Care Seeker or Support Worker clicked on profile button and their information was displayed. - Care Seeker or Support Worker clicked on edit button and updated his profile information.	
Expected Result	Care Seeker or Support Worker will be able to manage his profile.
Final Result	Care Seeker or Support Worker has managed his profile successfully.
Tested By	BC180404460

Job Post

Test Case Title	Job Post
Test Case ID	TC-04
Actor(s)	Care Seeker
Pre-Conditions	Care Seeker user must have an internet connection. Care Seeker must be logged in.
Task Sequence	
- Care seekers will log in to the web application through valid login credentials. - Care seeker will click on "Add Job" button in a menu on homepage for the purpose of posting a job and then user will give the information required in text fields and check boxes. - The application will validate the user information and then post a job with given information.	
Expected Result	User will be able to post a job.
Final Result	User has successfully posted a job.
Tested By	BC180404460

Message

Test Case Title	Message
Test Case ID	TC-05
Actor(s)	Care Seeker or Support Worker
Pre-Conditions	Care Seeker or Support Worker user must have an internet connection. Care Seeker or Support Worker must be logged in.
Task Sequence	
- Care Seeker will log in to the web application through valid login credentials. - Care Seeker will click on "Inbox" button and the Message will be displayed. - Care seeker can click on any Message button Then sent a reply to support worker. - Care seeker can also send message by opening on support worker profile then clicking in message button.	
Expected Result	Care Seeker or Support Worker will be able to send message.
Final Result	Care Seeker or Support Worker has successfully send message.
Tested By	BC180404460

Search Workers

Test Case Title	Search Workers
Test Case ID	TC-06
Actor(s)	Care Seeker
Pre-Conditions	Care Seeker user must have an internet connection. Care Seeker must be logged in.
Task Sequence	
- Care seeker will click on the "Find Worker" button in a menu on homepage for the purpose of finding a support worker and then user will give the information required in text fields and check boxes. - The application will validate the user information and then show a list of support workers.	
Expected Result	Care Seeker will be able search support worker.
Final Result	Care Seeker successfully search support worker
Tested By	BC180404460

Display Worker

Test Case Title	Display Workers
Test Case ID	TC-07
Actor(s)	Care Seeker
Pre-Conditions	Care Seeker user must have an internet connection. Care Seeker must be logged in.
Task Sequence	

- Care seeker will log in to the web application through valid login credentials.

- Care seeker will click on "Add Job" button in a menu on homepage for the purpose of posting a job and then user will give the information required in text fields and check boxes.
- The application will validate the user information and then post a job with given information.
- Now care seeker has to click on "Display Worker" button in order to see list of support worker available pre job requirement.

Expected Result	Care Seeker will be able see support worker.
Final Result	Care Seeker successfully see support worker
Tested By	BC180404460