Virtual University

Final Year Project
Care Seekers

Project Supervisor: Faizan Tahir

Submitted By
Group ID: S2302148F7
Name: Ali Bilal Ahmed
ID: BC180404460

Software Projects & Research Section,
Department of Computer Sciences,
Virtual University of Pakistan

Virtual University

CERTIFICATE

This is to certify that Ali Bilal Ahmed - BC180404460 have worked on and completed their Software Project at Software & Research Projects Section, Department of Computer Sciences, Virtual University of Pakistan in partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science under my guidance and supervision.

In our opinion, it is satisfactory and up to the mark and therefore fulfills the requirements of Bachelor of Science in Computer Science.

Supervisor / Internal Examiner

Faizan Tahir
Supervisor,
Software Projects & Research Section,
Department of Computer Sciences
Virtual University of Pakistan

____

External Examiner/Subject Specialist

____

Accepted By:

____

EXORDIUM

In the name of Allah, the Compassionate, the Merciful.

Praise be to Allah, Lord of Creation,
The Compassionate, the Merciful,
King of Judgment-day!

You alone we worship, and to You alone we pray for help,
Guide us to the straight path

The path of those who You have favored,

Not of those who have incurred Your wrath,
Nor of those who have gone astray.

DEDICATION

I dedicate this project to Allah, the Most Merciful, who has granted me the ability to program and develop this web application.

ACKNOWLEDGEMENT

I am grateful to my parents, esteemed teachers, and senior colleagues who provided unwavering support and encouragement throughout the completion of my final year project. I would also like to express my appreciation to the following individuals who played a significant role in assisting me:

- Sir Faizan Tahir
- Mohammed Uzair
- Tahir Nazir
- Usman Ahmed

Their invaluable contributions have made a lasting impact on the success of this project."

PREFACE

In the digital era, where connectivity and community play pivotal roles in our daily lives, "Care Connect" emerges as a transformative web-based platform tailored to the unique caregiving needs of individuals in Pakistan. This project stems from a profound understanding of the importance of fostering a supportive community that seamlessly connects Care Seekers with dedicated Support Workers.

The preface of "Care Connect" delves into the fundamental motivation behind the project – a commitment to providing accessible and personalized care services. In a society where the demand for specialized caregiving is diverse, this platform aims to bridge the gap, offering a user-friendly interface for both Care Seekers and Support Workers.

As we embark on this journey, the preface sets the stage for the project's ethos, emphasizing the significance of community, reliability, and tailored caregiving. "Care Connect" is not merely a website but a dynamic ecosystem designed to enhance the quality of life for individuals seeking assistance and those ready to provide it.

Table of Contents

CHAPTER - 1

Gathering & Analyzing Info

GATHERING & ANALYZING INFO

1.1 INTRODUCTION

In the world of today's challenges and possibilities, our project is like a shining light of new ideas. We are at the place where the needs of people and the cool things technology can do come together. This chapter is the start of our story it's about a website that wants to change how people find help, bringing together those who need care with those who provide it.

Life today has all sorts of needs, from taking care of older family members to helping with health issues and everyday tasks at home. Our project wants to be more than just a fix; it wants to be a way for people to connect and get the right help. By using the latest technology and being kind in how we help, we're creating a digital world where finding care isn't just about solving immediate problems but also about understanding and adapting to what people need.

As we go through the chapters ahead, each part is carefully planned to make sure we do things right. Our main goal is to bring together technology and kindness. It's not just about meeting the basic needs of our project; it's about making the experience better for both those who need care and those who offer it. Come with us on this adventure where technology and kindness meet, where every line of code shows our promise to a future where care isn't just a service but a shared journey.

1.2 PURPOSE

At the heart of our project lies a profound purpose—to revolutionize the landscape of care services. We are not merely constructing a website; we are architects of a digital sanctuary, where the pursuit of support transforms into an effortless and empowering odyssey.

Unveiling Our Purpose:

Facilitating Connections: Our purpose is rooted in the belief that genuine connections can be transformative. We strive to empower care seekers to effortlessly discover the support they seek while enabling support workers to engage meaningfully with those in need.

Simplifying Complexity: In a world often marred by intricacies, our purpose is to simplify. We are dedicated to streamlining the intricate process of locating, hiring, and coordinating support services, ensuring it becomes a straightforward, accessible, and effective endeavor.

Enriching Lives: Beyond the digital realm, our purpose extends to enriching lives. By fostering authentic connections and comprehending individual needs, we aspire to make a positive impact on both care seekers and support workers.

Essence of Our Purpose:

- **Innovation with Compassion:** Our purpose marries innovation with compassion. It is not just about technological advancements; it is about infusing every facet of our project with empathy and care.

- **Transformational Experiences:** We aspire to redefine the experience of seeking and providing care. It transcends mere transactions; it is a transformative journey, and our purpose is to imbue this journey with enrichment for all parties involved.

- **Community Building:** At its core, our purpose is to cultivate a community. A community where technology serves humanity, where connections are forged with compassion, and where care transcends the transactional—it becomes a shared endeavor.

Guiding Light:

Our purpose stands as the unwavering guide, ensuring that every line of code, every feature, and every interaction aligns with the overarching goal of making a positive and enduring impact in the realm of care services. This project is not a mere task; it is a purpose-driven undertaking to cultivate a world where care is not just accessible and personal but genuinely transformative.

1.3 PROJECT SCOPE

Defining the realm of possibilities and boundaries, the project scope is the compass guiding our endeavors. In straightforward terms, it outlines what our digital haven will encompass and, equally important, what it won't.

Embracing What We Cover:

Our focus is on crafting a user-friendly haven for those seeking support. Whether it's caring for elders, addressing health challenges, or managing day-to-day tasks at home, our platform strives to be the go-to destination. Care seekers will connect seamlessly with a community of dedicated support workers.

Blueprint of Inclusions:

User-Friendly Platform: Simplicity is key. Our website will be intuitive, ensuring a straightforward journey for users with no unnecessary complexities.

Support Worker Registration: Empowering those who want to help, the registration process will allow support workers to share vital details—experience, hourly rates, and references.

Job Posting for Care Seekers: Need help? Just post a job. Specify the service, share details, set a budget, and choose the time. It's that easy.

Service Categories: We're organizing services neatly. If you're looking for something specific, like baby care or cooking help, we've got categories to streamline the process.

Messaging Hub: Communication made simple. Care seekers and support workers can connect, discuss terms, and finalize details through our seamless messaging service.

Home Page Makeover: The heart of our platform—the home page—will be welcoming and efficient. Easy navigation with menus, visuals of services offered, and a central showcase of our care and support workers.

What's Not in the Blueprint:

Complexity Aversion: We're steering clear of unnecessary complications. Our goal is simplicity—no confusing elements.

Divergence from Purpose: Our focus is sharp. We won't venture into areas unrelated to our mission of connecting care seekers with support workers.

The Big Picture:

This project isn't just about coding; it's about creating a space where technology and compassion converge. It's not merely about getting tasks done; it's about building connections and enhancing the overall experience for everyone involved. Our scope is our commitment to executing this vision with precision and purpose.

1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

Navigating through our project requires a shared language. Here, we present key definitions, acronyms, and abbreviations that will serve as the linguistic foundation for our collective journey.

Key Definitions:

1. **Care Seekers:** Individuals actively seeking support services for elders, those facing health challenges, or requiring assistance with various domestic tasks.
2. **Support Workers:** Dedicated individuals offering their expertise in areas such as baby care, cooking, personal care, animal care, gym instruction, and domestic assistance.
3. **Bio Data:** Comprehensive information provided by support workers, encompassing details such as experience, hourly rates, and references.

Essential Acronyms:

1. **HTML**: HyperText Markup Language - the standard language for creating web pages and web applications.
2. **CSS**: Cascading Style Sheets - a style sheet language used for describing the look and formatting of a document written in HTML.
3. **JavaScript:** A programming language that enables interactive web pages.
4. **Laravel**: A full-stack PHP web application framework used for the development of robust and scalable web applications.

Abbreviations in Use:

1. **Gantt Chart:** A visual representation of a project schedule, outlining tasks against time.
2. **Chatify Package:** A communication package facilitating messaging services within the platform.

1.5 PROJECT REQUIREMENTS

This section forms the bedrock of our project, delineating the essential requirements that shape the core functionality and user experience of our web-based platform. Each requirement serves as a compass, guiding the development process to ensure our digital haven meets the diverse needs of care seekers and support workers seamlessly.

1.5.1 Functional Requirements

1. Support Worker Registration Process:

- Enable support workers to provide comprehensive bio data, showcasing their experience, hourly rates, and references.
- Implement a user-friendly interface for support workers to upload a profile picture, fostering a personalized connection.

2. Application Categories:

- Establish clear and distinct categories for various support worker specialties (e.g., Baby Care, Cooking), facilitating intuitive navigation for care seekers.

3. Job Posting (For Care Seekers):

- Empower care seekers to effortlessly post jobs with detailed information such as required services, service descriptions, addresses, estimated hourly budgets, and preferred service times.

4. Home Page Design:

- Craft an inviting home page with streamlined navigation menus (e.g., Post Job, Find Worker, Testimonials, About).
- Highlight images of all offered services prominently to provide an engaging visual experience.
- Feature a central section titled "Meet Our Care and Support Workers," showcasing images and concise information about each worker to foster a sense of familiarity.

1.5.2 Non-Functional Requirements

1. Usability:

Ensure the website's design promotes ease of use, catering to a diverse audience and fostering an inclusive user experience.

2. Performance:

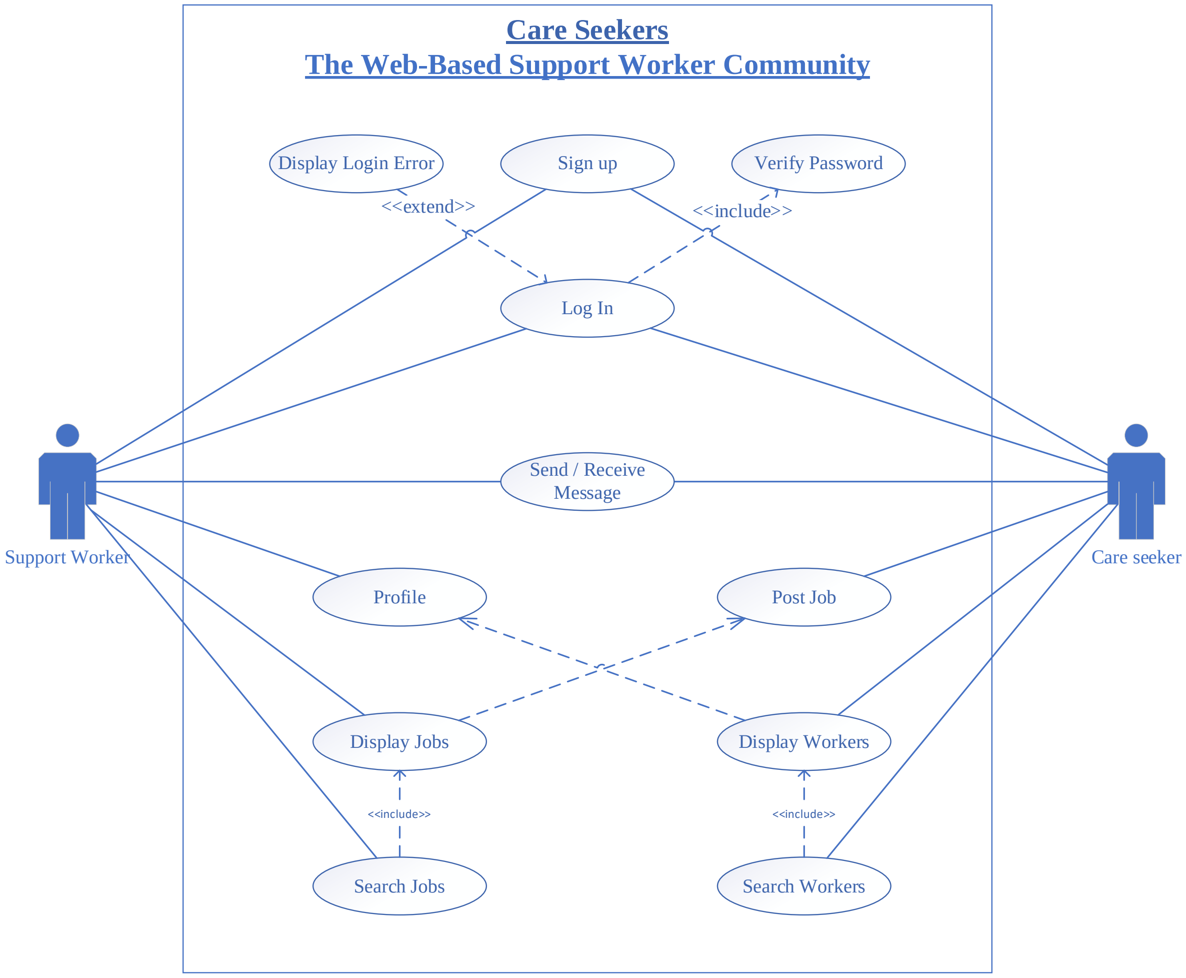
Optimize platform performance to handle concurrent user interactions efficiently, ensuring a responsive and seamless user interface.

3. Security:

Implement robust security measures to safeguard user data, fostering trust and maintaining stringent privacy standards.

1.6 USE CASES AND USAGE SCENARIOS

1.6.1 Use Case Diagrams



USE CASE DESCRIPTIONS

1. **Sign Up:**
- **Actor:** Care Seeker and Support Worker
- **Description:** Both care seekers and support workers can sign up on the platform by providing necessary details.
2. **Log In:**
- **Actor:** Care Seeker and Support Worker
- **Description:** Users log in to their accounts; includes an extend relationship with "Display Login Error" and an include relationship with "Verify Password."
3. **Send/Receive Message:**
- **Actor:** Care Seeker and Support Worker
- **Description:** Both care seekers and support workers can engage in a messaging system to communicate about job details and requirements.
4. **Post Job:**
- **Actor:** Care Seeker

- **Description:** Care seekers can post job listings, specifying details like required services, descriptions, and estimated budgets.

5. **Display Workers:**

- **Actor:** Care Seeker
- **Description:** Care seekers can view and explore profiles of available support workers.

6. **Search Workers:**

- **Actor:** Care Seeker
- **Description:** Care seekers can search for support workers based on criteria and view the results; includes a relationship with "Display Workers."

7. **Display Jobs**:

- **Actor:** Support Worker
- **Description:** Support workers can view posted job listings.

8. **Search Jobs:**

- **Actor:** Support Worker
- **Description:** Support workers can search for jobs based on criteria and view the results; includes a relationship with "Display Jobs."

9. **Profile:**

- **Actor:** Support Worker
- **Description:** Support workers can access and manage their profiles on the platform.

1.6.2 Usage Scenarios

Sign up

Use Case Title	Sign up
Use Case ID	UC-001
Actor(s)	Care Seeker and Support Worker
Description	Actors, both care seekers and support workers, sign up on the platform to create their accounts and provide essential details.
Pre-Conditions	The actors have internet access, and the platform sign-up page is accessible.
Task Sequence	1. Actors access the sign-up page. 2. Actors provide necessary information (e.g., username, email, password, bio data). 3. Actors submit the sign-up form.
Exception	• Duplicate Username Exception • Duplicate Email Exception
Post Conditions	Actors have successfully created accounts and can log in.
Alternative Path	If the provided username or email is already in use, actors receive an error message and are prompted to choose different credentials.
Author (s)	BC180404460

Log In

Use Case Title	Log In
Use Case ID	UC-002
Actor(s)	Care Seeker and Support Worker
Description	Actors, both care seekers and support workers, log in to their accounts to access platform features.
Pre-Conditions	Actors have registered accounts on the platform.
Task Sequence	1. Actors access the login page. 2. Actors provide their username and password. 3. Actors submit the login form.
Exception	• Login Error Exception • Invalid Credentials Exception
Post Conditions	Actors are successfully logged in and can access their respective dashboards.
Alternative Path	If the provided credentials are incorrect, actors receive a login error message.
Author (s)	BC180404460

Send/Receive Message

Use Case Title	Send/Receive Message
Use Case ID	UC-003
Actor(s)	Care Seeker and Support Worker
Description	Actors engage in a messaging system to communicate about job details and requirements.
Pre-Conditions	Actors are logged in to their accounts and have active conversations.
Task Sequence	1. Actors navigate to the messaging section. 2. Actors select a conversation or initiate a new one. 3. Actors compose and send messages. 4. Actors receive and read messages.
Exception	• Message Sending Exception • Message Receiving Exception
Post Conditions	Messages are successfully sent and received.
Alternative Path	
Author (s)	BC180404460

Post Job

Use Case Title	Post Job
Use Case ID	UC-004
Actor(s)	Care Seeker
Description	Care seekers post job listings to find suitable support workers.
Pre-Conditions	Care seeker is logged in and has a job to post.
Task Sequence	1. Care seeker accesses the "Post Job" section. 2. Care seeker provides job details, including required services, descriptions, address, estimated budgets, and preferred service times. 3. Care seeker submits the job posting.
Exception	• Job Posting Exception
Post Conditions	Job posting is successfully added to the platform.
Alternative Path	Care seeker can edit or delete a posted job.
Author (s)	BC180404460

Display Workers

Use Case Title	Display Workers
Use Case ID	UC-005
Actor(s)	Care Seeker
Description	Care seekers explore and view profiles of support workers on the platform.
Pre-Conditions	are seeker is logged in and is actively searching for support workers.
Task Sequence	1. Care seeker navigates to the "Display Workers" section. 2. Care seeker views profiles of available support workers.
Exception	None
Post Conditions	Care seeker gains insights into available support workers.
Alternative Path	None
Author (s)	BC180404460

Sign up

Use Case Title	Search Workers
Use Case ID	UC-006
Actor(s)	Care Seeker
Description	Care seekers search for support workers based on specific criteria.
Pre-Conditions	Care seeker receives relevant search results.
Task Sequence	1. Care seeker accesses the "Search Workers" section. 2. Care seeker defines search criteria (e.g., service category, experience, ratings). 3. Care seeker views search results.
Exception	• Search Results Not Found Exception
Post Conditions	Care seeker is logged in and actively seeking support workers.
Alternative Path	Care seeker can refine or broaden search criteria.
Author (s)	BC180404460

Display Jobs

Use Case Title	Display Jobs
Use Case ID	UC-007
Actor(s)	Support Worker
Description	Support workers explore and view job listings posted by care seekers.
Pre-Conditions	Support worker is logged in and actively searching for job opportunities.
Task Sequence	1. Support worker navigates to the "Display Jobs" section. 2. Support worker views posted job listings.
Exception	None
Post Conditions	Support worker gains insights into available job opportunities.
Alternative Path	None
Author (s)	BC180404460

Search Jobs

Use Case Title	Search Jobs
Use Case ID	UC-008
Actor(s)	Support Worker
Description	Support workers search for jobs based on specific criteria.
Pre-Conditions	Support worker is logged in and actively seeking job opportunities.
Task Sequence	1. Support worker accesses the "Search Jobs" section. 2. Support worker defines search criteria (e.g., service category, budget, location). 3. Support worker views search results.
Exception	- Search Results Not Found Exception
Post Conditions	Support worker receives relevant search results.
Alternative Path	Support worker can refine or broaden search criteria.
Author (s)	BC180404460

Profile

Use Case Title	Profile
Use Case ID	UC-009
Actor(s)	Support Worker
Description	Support workers access and maintain their profiles on the platform.
Pre-Conditions	Support worker is logged in and wishes to manage their profile.
Task Sequence	1. Support worker accesses the "Profile" section. 2. Support worker views and manages their profile information.
Exception	- Profile Update Exception
Post Conditions	Support worker successfully views and updates profile details.
Alternative Path	Support worker can edit or update profile information.
Author (s)	BC180404460

1.7 DEVELOPMENT METHODOLOGY

This section articulates our selection of the waterfall development methodology and provides a robust rationale for this strategic choice.

1.7.1 Chosen Methodology

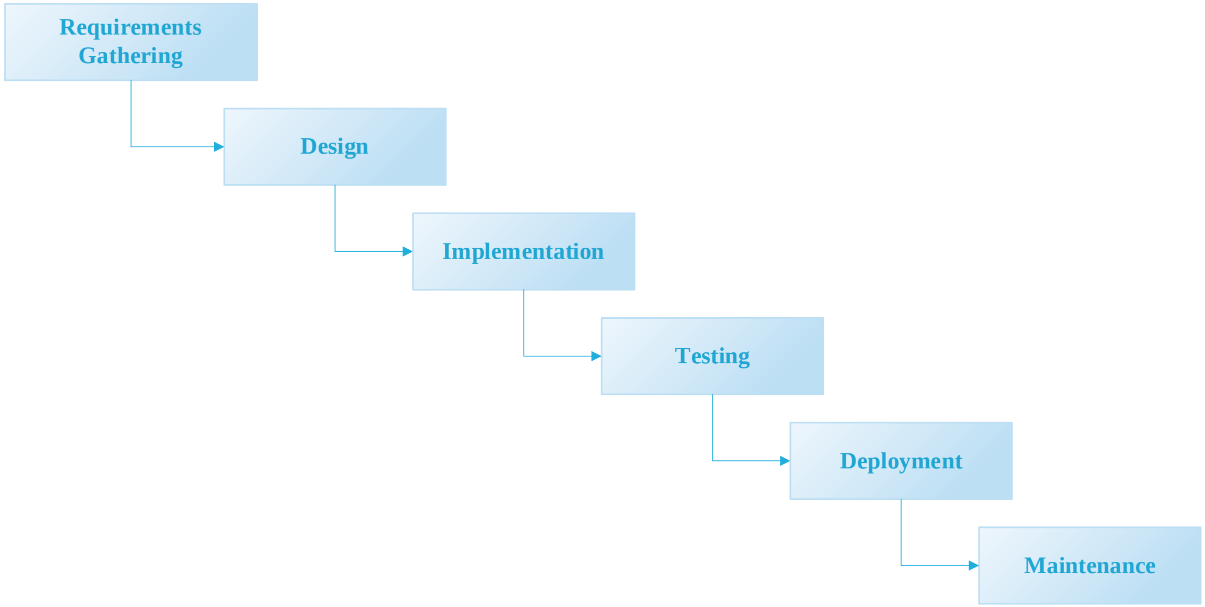
Waterfall Development Methodology

Overview:

- **Structured Progression:** The waterfall model signifies a meticulously planned and sequential approach to project development, with distinct phases unfolding in a predetermined order.
- **Phased Milestones:** Each project phase, from requirements gathering to deployment, acts as a milestone, offering a clear delineation of progress and achievements.
- **Rigorous Validation:** A hallmark of the waterfall methodology is its insistence on completing each phase before moving forward, facilitating thorough validation and verification processes.

Description: The application of the waterfall methodology in our project involves a step-by-step progression through well-defined phases, ensuring a comprehensive development process tailored to the unique requirements of our web-based platform.

The Waterfall Model



Project Phases:

1. **Requirements Gathering:**

- **Objective:** Thoroughly understand the needs of care seekers and support workers.
- **Actions:** Engage stakeholders, conduct interviews, and document detailed project requirements.
- **Outcome:** A comprehensive and well-documented set of requirements that forms the foundation for subsequent phases.

2. **Design:**
   - **Objective:** Develop a blueprint for the user interface and overall system architecture.
   - **Actions:** Create wireframes, design the user experience, and define the system's structure.
   - **Outcome:** Detailed design documents and prototypes that guide the development team in building the platform.

3. **Implementation:**
   - **Objective:** Translate design specifications into actual code and develop the web-based platform.
   - **Actions:** Write and test the code, integrate features, and build the core functionalities.
   - **Outcome:** A functional and executable version of the platform.

4. **Testing:**
   - **Objective:** Verify that the developed platform meets specified requirements and functions as intended.
   - **Actions:** Conduct thorough testing, including unit testing, integration testing, and user acceptance testing.
   - **Outcome**: A rigorously validated platform with identified and addressed issues.

5. **Deployment:**
   - **Objective:** Release the platform to the production environment for public access.
   - **Actions:** Configure servers, migrate databases, and ensure a smooth transition from the development environment.
   - **Outcome:** The web-based platform is live and accessible to users.

6. **Maintenance and Support:**
   - **Objective:** Address post-deployment issues, provide ongoing support, and implement necessary updates.
   - **Actions:** Monitor platform performance, address user feedback, and make improvements as needed.
   - **Outcome:** A stable and evolving platform that continues to meet user needs.

1.7.2 Reasons for Chosen Methodology

Clear and Stable Requirements Tailored for Care Seeker Precision:
The waterfall model stands out as the ideal choice for projects where requirements are not just clear but also need to be tailored with precision. In our case, where the needs and expectations of care seekers are paramount, the structured nature of the waterfall approach ensures that each element is defined with meticulous attention, guaranteeing a solution finely tuned to the care seeker's journey.

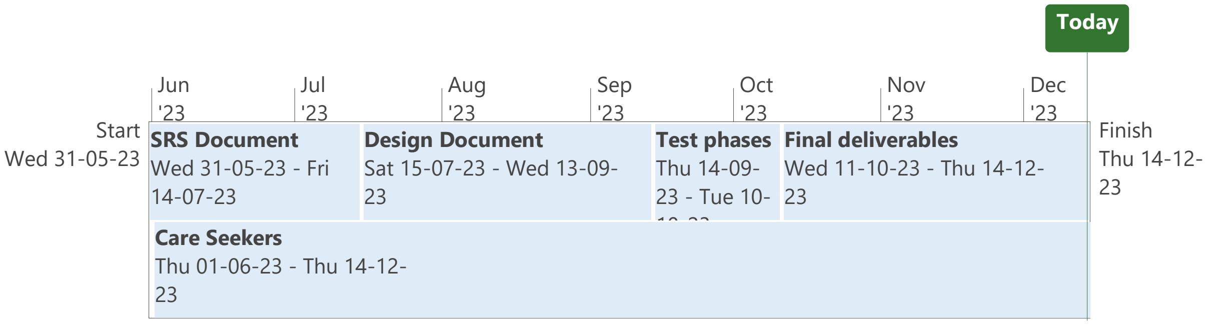
Sequential Progression Aligned with Care Seeker Journey:
The step-by-step progression of the waterfall model mirrors the careful and sequential steps a care seeker takes in finding the right support. This approach ensures that each phase is a deliberate move toward refining the platform to cater to the unique requirements of care seekers. It’s not just a development methodology; it's a thoughtful alignment with the care seeker's journey.

Thorough Documentation Enhancing Future Care Clarity:
In the realm of care-seeking, where details matter, the emphasis on comprehensive documentation in the waterfall methodology is not just a procedural requirement but a strategic asset. Each documented phase becomes a guide, ensuring that the nuances of care-seeking processes are preserved for future enhancements and optimizations. This commitment to documentation aligns seamlessly with the future clarity needs of care seekers.

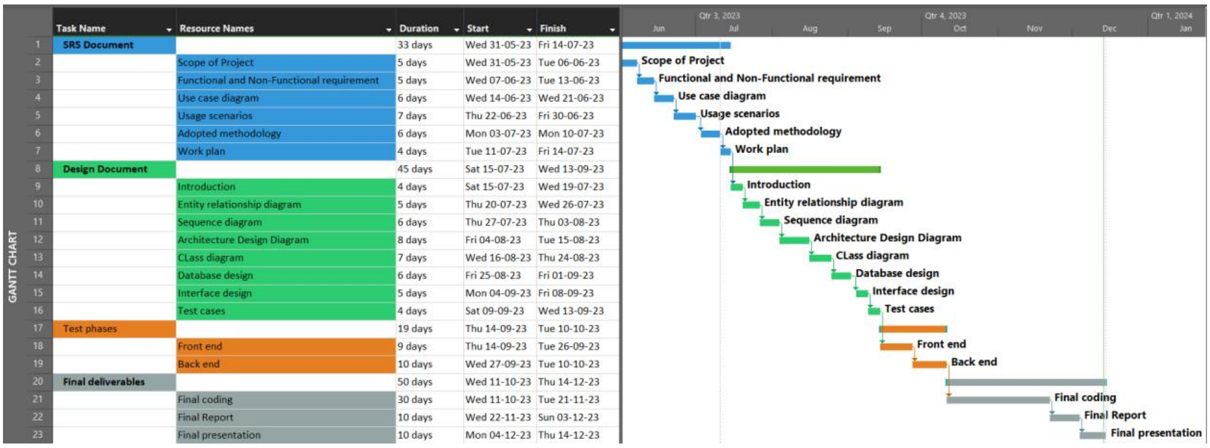
By adopting the waterfall methodology, our project acknowledges the unique precision required for the care seeker experience. This choice ensures that our development journey isn't just a series of phases but a thoughtful progression, resonating with the care seeker's meticulous approach to finding the right support.

1.7.2 Work Plan (Gantt Chart)

Timeline



Gantt Chart



	Task Name	Resource Names	Duration	Start	Finish
1	SRS Document		33 days	Wed 31-05-23	Fri 14-07-23
2		Scope of Project	5 days	Wed 31-05-23	Tue 06-06-23
3		Functional and Non-Functional requirement	5 days	Wed 07-06-23	Tue 13-06-23
4		Use case diagram	6 days	Wed 14-06-23	Wed 21-06-23
5		Usage scenarios	7 days	Thu 22-06-23	Fri 30-06-23
6		Adopted methodology	6 days	Mon 03-07-23	Mon 10-07-23
7		Work plan	4 days	Tue 11-07-23	Fri 14-07-23
8	Design Document		45 days	Sat 15-07-23	Wed 13-09-23
9		Introduction	4 days	Sat 15-07-23	Wed 19-07-23
10		Entity relationship diagram	5 days	Thu 20-07-23	Wed 26-07-23
11		Sequence diagram	6 days	Thu 27-07-23	Thu 03-08-23
12		Architecture Design Diagram	8 days	Fri 04-08-23	Tue 15-08-23
13		CLass diagram	7 days	Wed 16-08-23	Thu 24-08-23
14		Database design	6 days	Fri 25-08-23	Fri 01-09-23
15		Interface design	5 days	Mon 04-09-23	Fri 08-09-23
16		Test cases	4 days	Sat 09-09-23	Wed 13-09-23
17	Test phases		19 days	Thu 14-09-23	Tue 10-10-23
18		Front end	9 days	Thu 14-09-23	Tue 26-09-23
19		Back end	10 days	Wed 27-09-23	Tue 10-10-23
20	Final deliverables		50 days	Wed 11-10-23	Thu 14-12-23
21		Final coding	30 days	Wed 11-10-23	Tue 21-11-23
22		Final Report	10 days	Wed 22-11-23	Sun 03-12-23
23		Final presentation	10 days	Mon 04-12-23	Thu 14-12-23

1.7.2 Project Schedule (Submission Calendar)

Our project unfolds on a carefully orchestrated schedule, ensuring each milestone is met with precision and diligence. The timeline, encapsulated in our Submission Calendar, reflects a strategic cadence that navigates through crucial project phases.

SRS Document (Submission Date: July 14, 2023):
Commencement: May 31, 2023
Conclusion: July 14, 2023

Design Document (Submission Date: September 13, 2023):
Commencement: July 15, 2023
Conclusion: September 13, 2023

Prototype (Submission Date: October 10, 2023):
Commencement: September 14, 2023
Conclusion: October 10, 2023

Final Deliverable (Submission Date: December 14, 2023):
Commencement: October 11, 2023
Conclusion: December 14, 2023

CHAPTER 2

Designing the Project

Designing the Project

2.1 Introduction

In the realm of caregiving, our web-based project is designed to bridge the gap between care seekers and dedicated support workers. Tailored for individuals seeking assistance with tasks ranging from elder care to cooking and personal well-being, our platform creates a vibrant online community. Care seekers can effortlessly post job requirements or explore profiles within our support worker network. Support workers, in turn, register on the website, offering their expertise and availability. The interaction unfolds seamlessly, as care seekers post job details, and support workers respond through an integrated messaging service. Upon mutual agreement, support workers formally accept the job. This introduction sets the stage for a compassionate and efficient platform that empowers care seekers to connect with the right support, ensuring a harmonious blend of technology and caregiving.

2.2 Purpose

The purpose of our project is to establish a user-centric, web-based platform that harmoniously connects care seekers with dedicated support workers. Rooted in the commitment to simplify and enhance the caregiving process, our platform aims to provide a seamless and intuitive experience for both care seekers and support workers. For care seekers, the purpose is to facilitate the effortless discovery of qualified support workers who align with their specific needs. Simultaneously, for support workers, the purpose is to offer a user-friendly registration process and a responsive platform to connect with care seekers. By fostering a community-driven approach, our project aspires to redefine the way caregiving services are sought and delivered, ensuring a purposeful and empathetic digital space for both parties involved.

2.3 Scope

Within the expansive scope of our project, we aim to create an inclusive and accessible web-based platform that caters to the diverse needs of care seekers and support workers. The scope encompasses a comprehensive range of caregiving services, including but not limited to elder care, assistance for the sick, baby care, cooking, personal care, animal care, gym instruction, and domestic assistance. Care seekers will be empowered to effortlessly post job requirements or explore the profiles of registered support workers, initiating a seamless connection process. Support workers, on the other hand, will find a purposeful space to showcase their skills and respond to job postings through an integrated messaging service. The project's scope extends to the design of a user-friendly homepage, featuring essential menus (e.g., Post Job, Find Worker) and showcasing the diverse services offered. In essence, our project ambitiously seeks to redefine the landscape of caregiving by providing a digital platform that is both comprehensive and user-centric.

2.4 Definitions, Acronyms, and Abbreviations

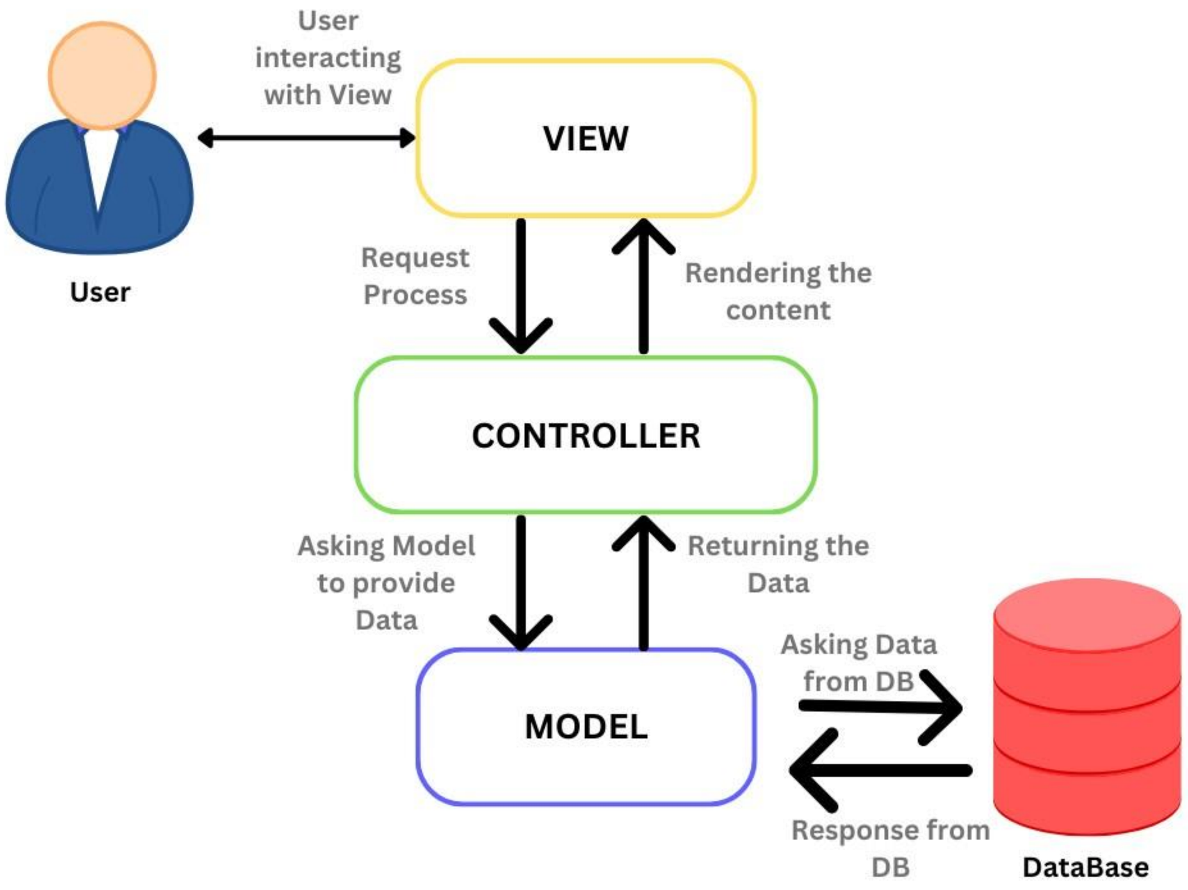
To ensure clear communication throughout our project documentation, the following definitions, acronyms, and abbreviations are established:

1. **MVC:** Model-View-Controller - An architectural pattern widely used in web development for organizing code into three interconnected components: Model (handles data and database operations), View (handles user interface and presentation logic), and Controller (manages user input, processes requests, and interacts with the Model and View).
2. **SRS:** Software Requirements Specification - A document that outlines the functional and non-functional requirements of a software system.
3. **GUI:** Graphical User Interface - The visual elements through which users interact with a software application.
4. **HTML:** Hypertext Markup Language - A standard markup language for creating web pages.
5. **CSS:** Cascading Style Sheets - A style sheet language used for describing the presentation of a document written in HTML.
6. **JavaScript:** A programming language that enables interactive and dynamic web pages.
7. **Laravel:** A PHP web application framework that follows the MVC pattern, offering tools and conventions for building scalable and maintainable applications.
8. **API:** Application Programming Interface - A set of rules that allows one software application to interact with another.
9. **UX:** User Experience - The overall experience a user has when interacting with a system, including aspects of usability, accessibility, and satisfaction.
10. **UI**: User Interface - The visual elements and controls through which users interact with a software application.
11. **SQL:** Structured Query Language - A domain-specific language used for managing and manipulating relational databases.
12. **Gantt Chart:** A visual representation of project timelines and tasks, often used in project management.
13. **KB:** Kilobytes - A unit of digital information storage.
14. **MB:** Megabytes - A unit of digital information storage.
15. **GB:** Gigabytes - A unit of digital information storage.

2.5 ARCHITECTURAL REPRESENTATION (ARCHITECTURE DIAGRAM)

Model-View-Controller (MVC) Architecture

In our caregiving platform, we leverage the power of the Model-View-Controller (MVC) architectural pattern to create a robust and well-organized structure that aligns with Laravel's conventions.



Model (M):

Entities Representation:
We define models to represent key entities such as users, support workers, and job listings. Each model encapsulates the data structure and interacts with the database through Laravel's Eloquent ORM, ensuring efficient data retrieval and storage.
Database Operations:
Models handle database operations, including querying, inserting, updating, and deleting records. This ensures a seamless interaction with the underlying database, promoting data integrity and consistency.

View (V):

Blade Templates:
Laravel's Blade templating engine empowers our views, allowing us to create dynamic and visually appealing user interfaces. Blade templates enable us to structure and present information in a user-friendly manner, fostering an intuitive experience for care seekers and support workers.
Isolation of Presentation Logic:
Views focus solely on presentation logic, separating it from the application's core business logic. This isolation enhances maintainability and facilitates the adaptation of different layouts and designs.

Controller (C):

Request Handling:
Controllers act as the bridge between models and views, handling user requests and orchestrating the flow of data. Laravel's routing system directs HTTP requests to the appropriate controller methods, ensuring a well-organized and structured response.
Business Logic:
Business logic resides in controllers, where we process input, interact with models to fetch or update data, and determine the appropriate response. This separation of concerns enhances code readability and facilitates easier updates or modifications.

Overall Benefits:

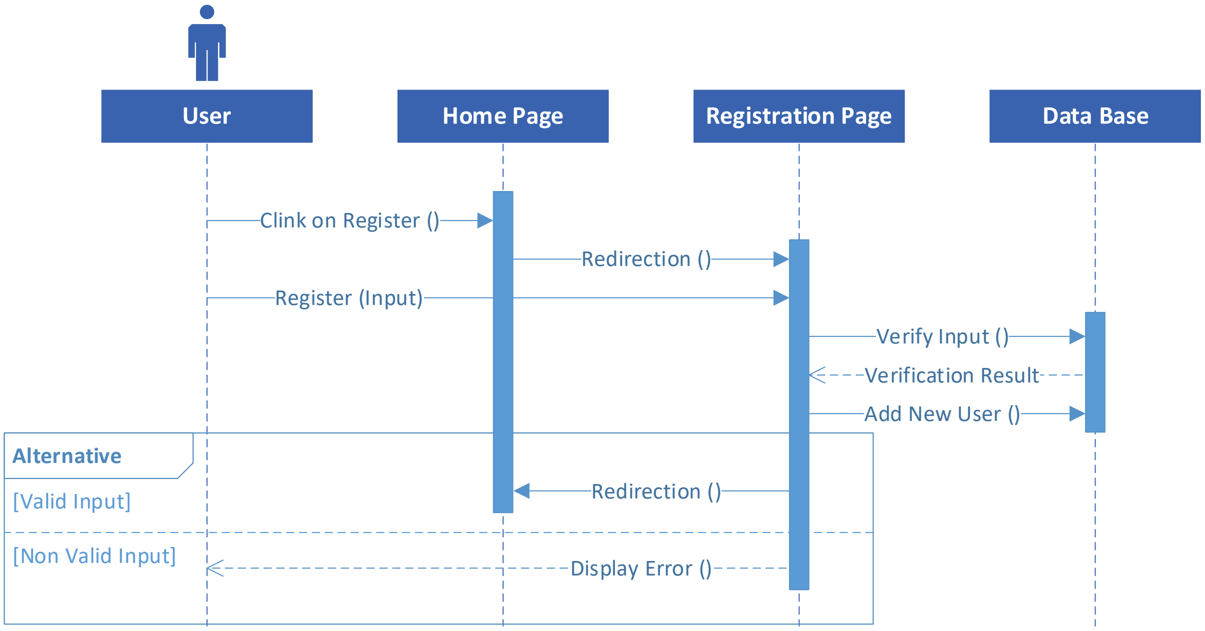
1. **Organization and Maintainability:**
   The MVC pattern provides a clear separation of concerns, making it easier to organize code and maintain different aspects of the application independently.
2. **Scalability:**
   The modular structure of MVC allows for scalability, enabling us to add new features or make modifications without disrupting the entire system.
3. **Developer Productivity:**
   MVC enhances developer productivity by providing a clear structure and conventions, allowing team members to work on different components simultaneously.

By embracing the MVC architecture within Laravel, our caregiving platform is poised for success, offering a reliable, scalable, and maintainable solution for care seekers and support workers alike.

2.6 DYNAMIC MODEL: SEQUENCE DIAGRAMS

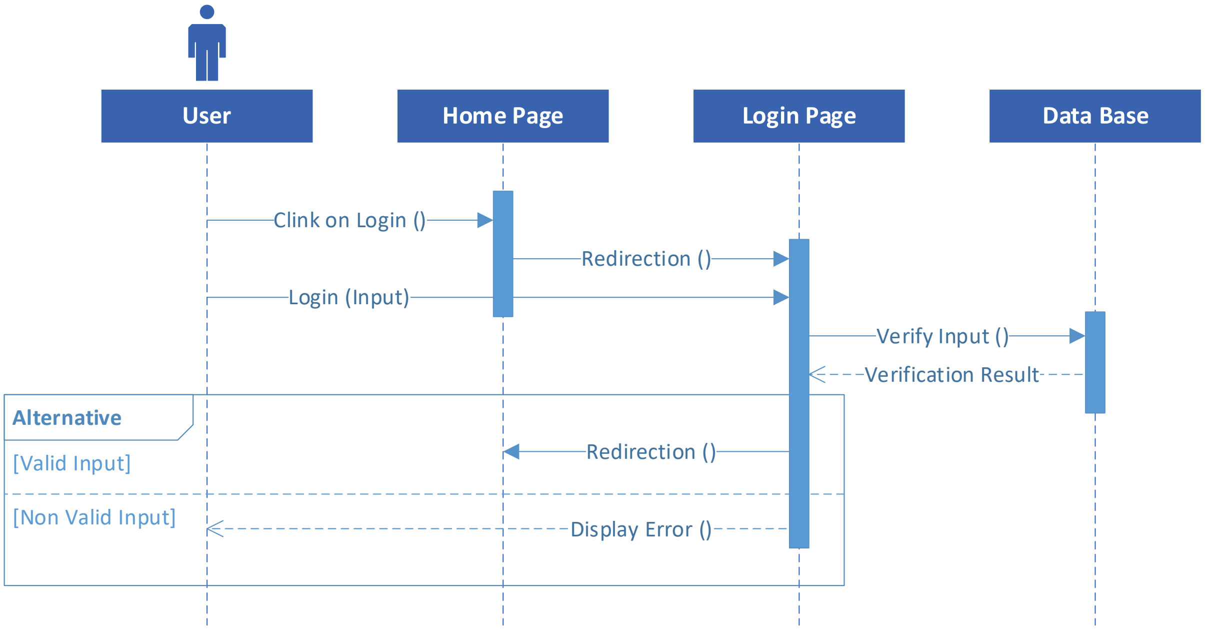
Sign Up

Sequence Diagrams of Sign Up



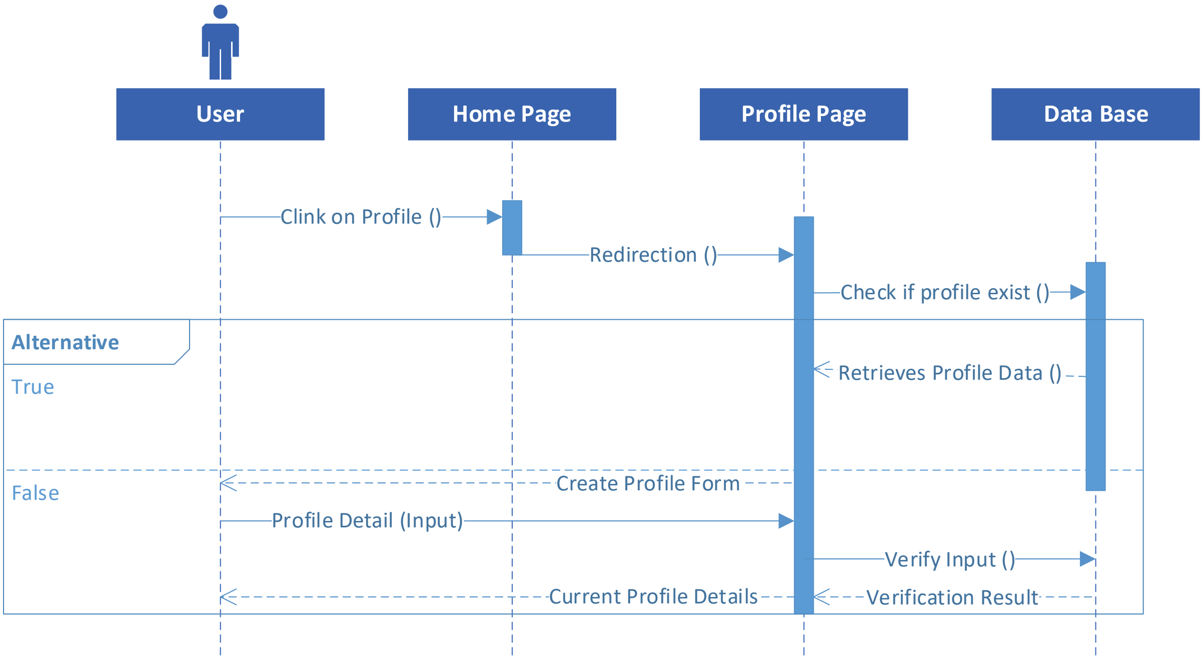
Login

Sequence Diagrams of Log In



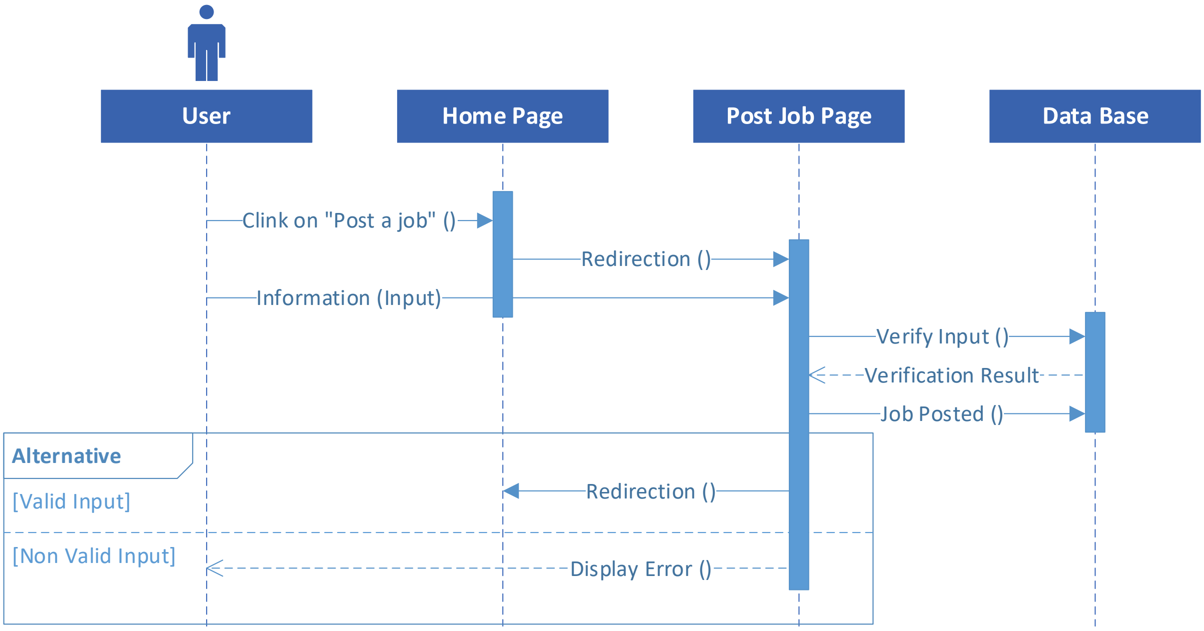
Profile

Sequence Diagrams of Profile



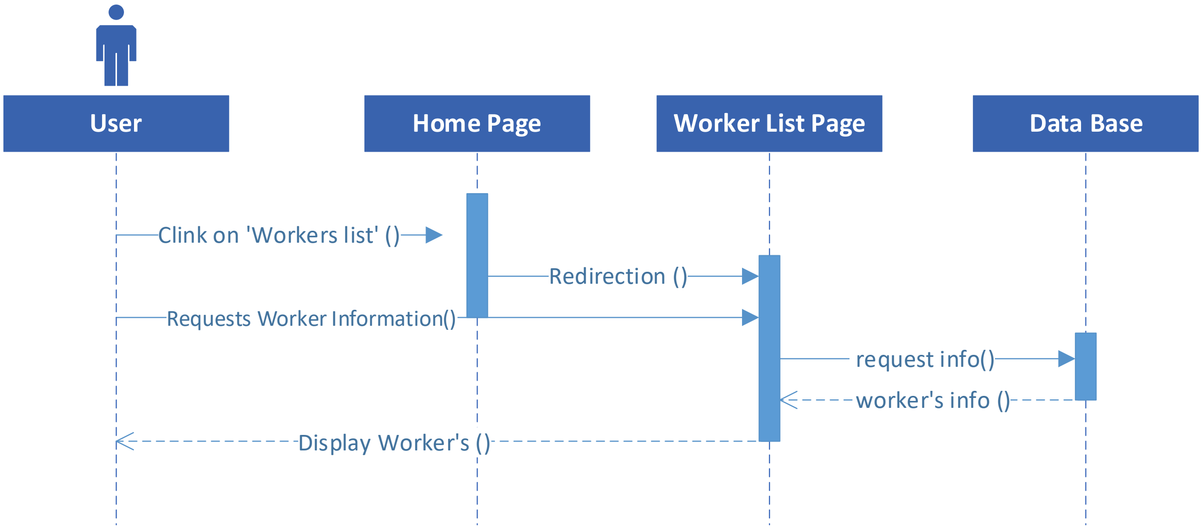
Posting Job

Sequence Diagrams of Posting Job



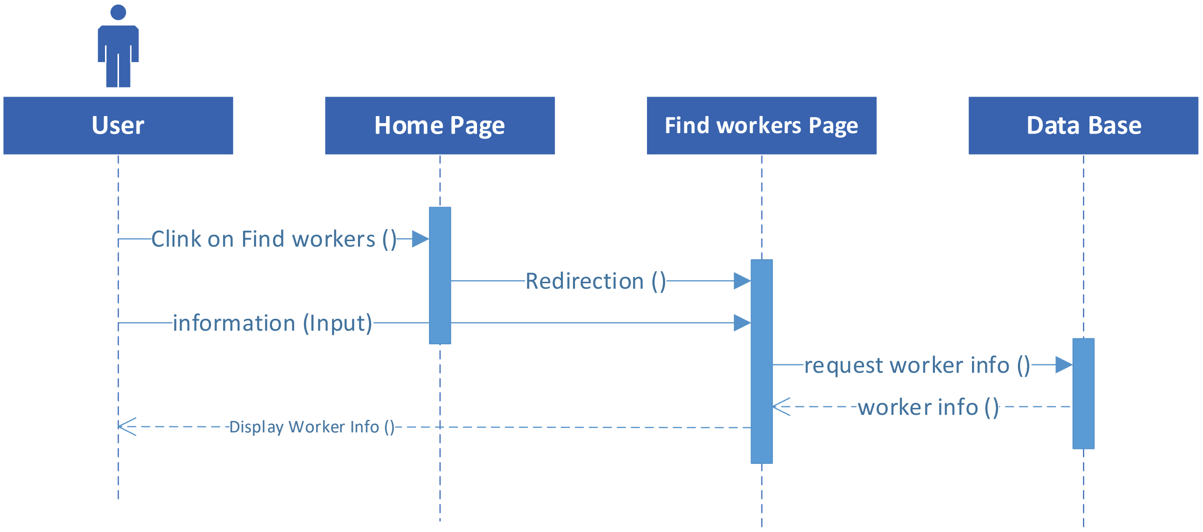
Worker list

Sequence Diagrams of Display Worker



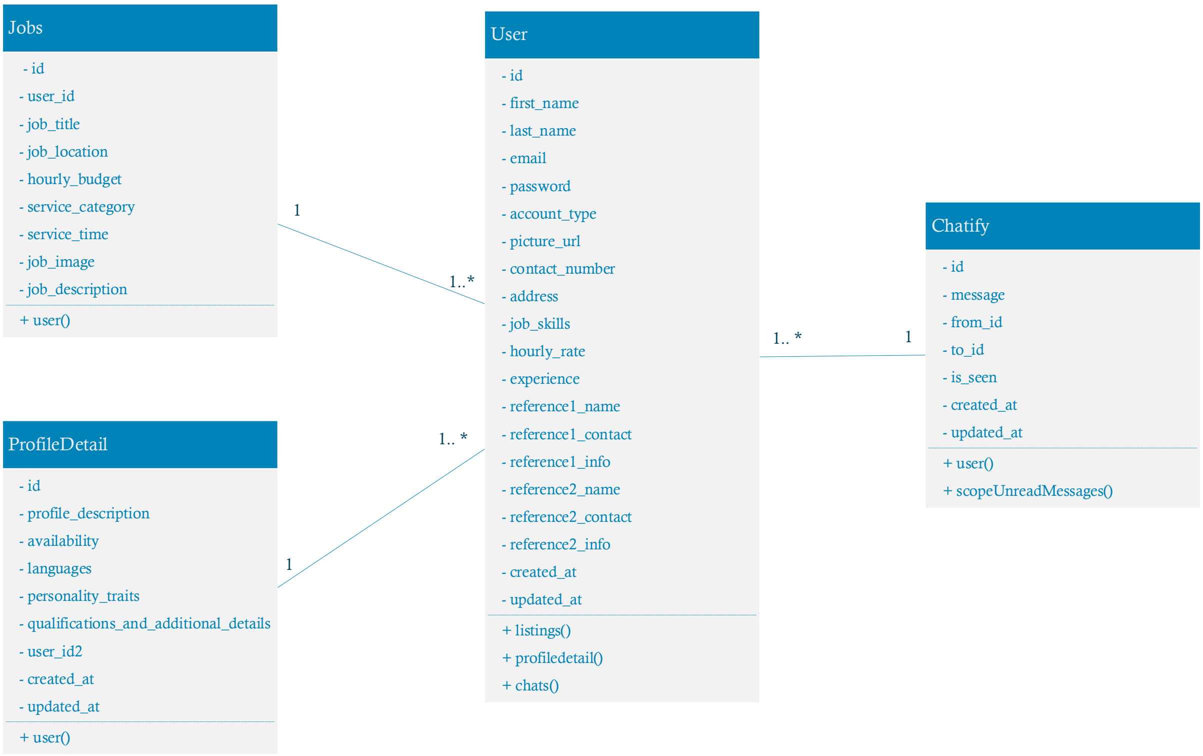
Search Worker

Sequence Diagrams of Search Worker



2.7 OBJECT MODEL/LOGICAL MODEL: CLASS DIAGRAM

Class Diagram of Project Care Seeker



Explanation:

jobs Class:

- **id:** An identifier for each job.
- **user_id:** Foreign key linking to the User class, representing the owner of the job.
- job_title, job_location, hourly_budget, service_category, service_time, job_image, **job_description:** Properties related to the details of a job.
- created_at, updated_at: Timestamps for when the job was created or last updated.

User Class:

- **id:** An identifier for each user.
- first_name, last_name, email, password, account_type, picture_url, contact_number, address, job_skills, hourly_rate, experience, reference1_name, reference1_contact, reference1_info, reference2_name, reference2_contact, reference2_info: Properties related to user information.
- **listings:** Relationship indicating a one-to-many connection to the jobs class.
- **profiledetail:** Relationship indicating a one-to-one connection to the ProfileDetail class.
- **created_at, updated_at:** Timestamps for when the user was created or last updated.

ProfileDetail Class:

- **id:** An identifier for each profile detail.
- profile_description, availability, languages, personality_traits, qualifications_and_additional_details: Properties related to the profile of a user.
- user_id2: Foreign key linking to the User class, representing the owner of the profile detail.
- **created_at, updated_at:** Timestamps for when the profile detail was created or last updated.

Chatify Class:

- **id**: An identifier for each chat.
- **user_id:** Foreign key linking to the User class, representing the sender of the message.
- **friend_id**: Foreign key linking to the User class, representing the receiver of the message.
- **message, type, status:** Properties related to the chat message details.
- **created_at, updated_at:** Timestamps for when the chat message was created or last updated.

Relationships:

There's a one-to-many relationship between User and jobs, meaning a user can have multiple jobs.

There's a one-to-one relationship between User and ProfileDetail, meaning each user has one associated profile detail.

There's a one-to-many relationship between User and Chatify, indicating a user can have multiple chat messages.

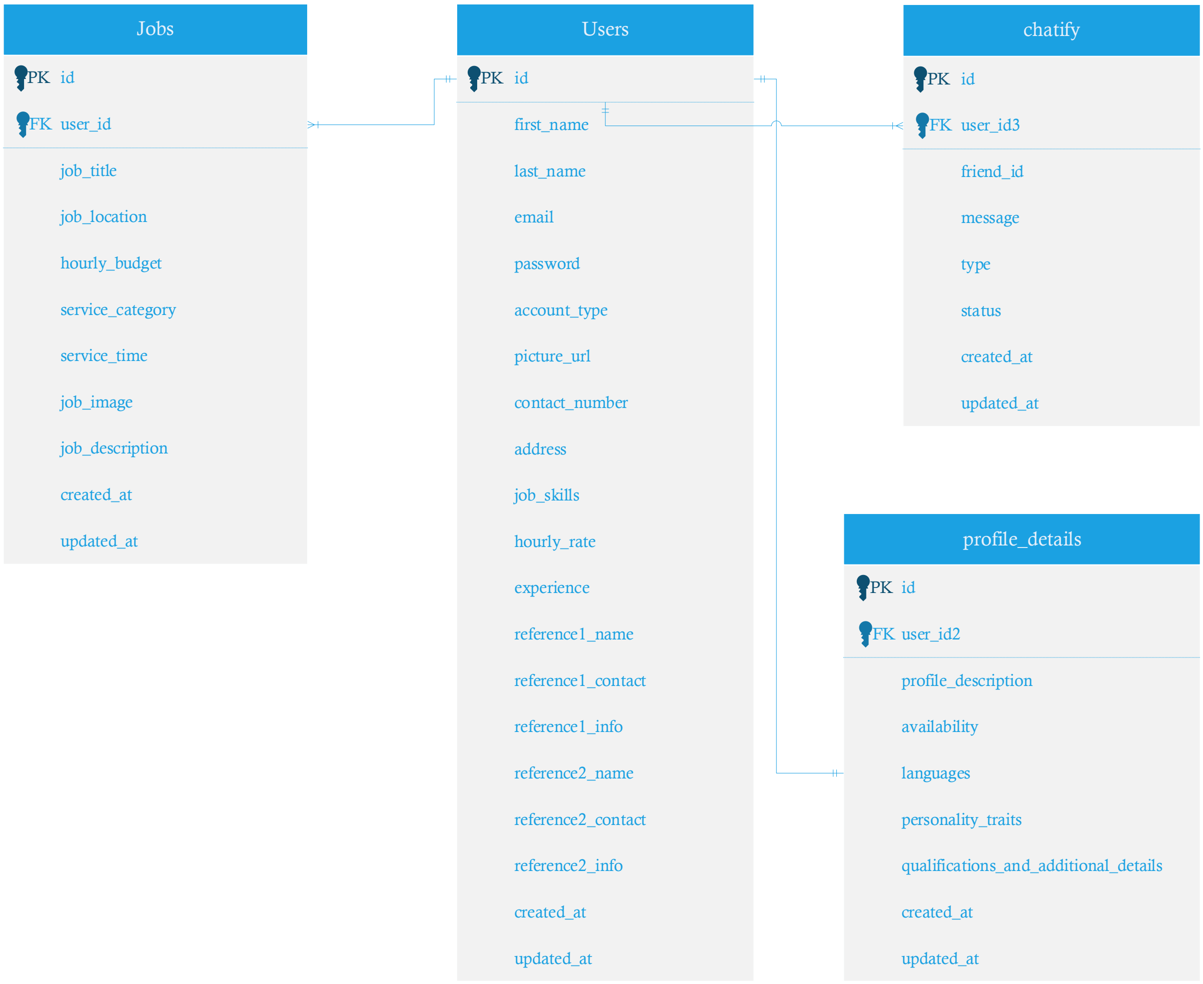
Additional Notes:

The diagram provides a clear overview of how classes are related and their respective properties.

The timestamps (created_at and updated_at) are part of Laravel's Eloquent model conventions and are automatically managed by the framework.

2.8 DATABASE MODEL (DATABASE DIAGRAM)

Database Design Diagram



Tables:

1. **users:**

- id (Primary Key): Unique identifier for each user.
- first_name: First name of the user.
- last_name: Last name of the user.
- email: Email address of the user.
- password: Encrypted password for user authentication.
- account_type: Type of account (e.g., CareSeeker or SupportWorker).
- picture_url: URL/path to the user's profile picture.
- contact_number: Contact number of the user.
- address: Address of the user.
- job_skills: Skills associated with the user (specific to SupportWorkers).
- hourly_rate: Hourly rate charged by the SupportWorker.
- experience: Work experience details of the SupportWorker.
- reference1_name: Name of the first reference for the SupportWorker.

- reference1_contact: Contact information of the first reference.
- reference1_info: Additional information about the first reference.
- reference2_name: Name of the second reference for the SupportWorker.
- reference2_contact: Contact information of the second reference.
- reference2_info: Additional information about the second reference.
- remember_token: Token for "remember me" functionality.
- created_at: Timestamp for record creation.
- updated_at: Timestamp for last update.

2. **jobs:**

- id (Primary Key): Unique identifier for each job.
- user_id (Foreign Key): Reference to the user who posted the job.
- job_title: Title of the job.
- job_location: Location where the job is located.
- hourly_budget: Budget allocated for the job on an hourly basis.
- service_category: Category of service required for the job.
- service_time: Time details associated with the job.
- job_image: URL/path to an image related to the job.
- job_description: Detailed description of the job.
- created_at: Timestamp for record creation.
- updated_at: Timestamp for last update.

3. **profile_details:**

- id (Primary Key): Unique identifier for each profile detail.
- profile_description: Description of the user's profile.
- availability: Availability details of the user.
- languages: Languages spoken by the user.
- personality_traits: Traits describing the user's personality.
- qualifications_and_additional_details: Additional qualifications or details.
- user_id2 (Foreign Key): Reference to the user associated with the profile.
- created_at: Timestamp for record creation.
- updated_at: Timestamp for last update.

4. **chatify:**

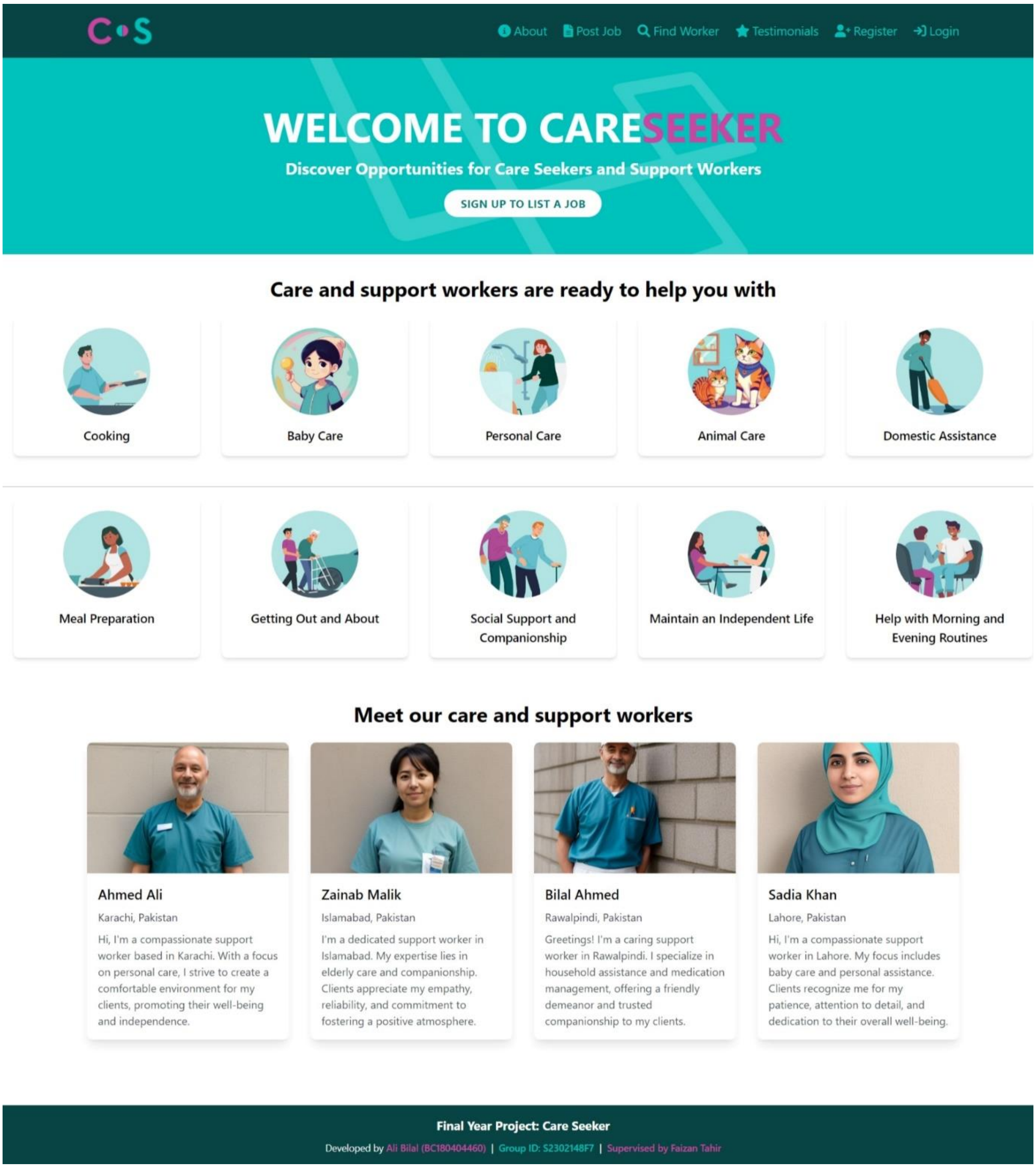
- id (Primary Key): Unique identifier for each chat message.
- user_id (Foreign Key): Reference to the user sending the message.
- friend_id (Foreign Key): Reference to the user receiving the message.
- message: Content of the chat message.
- type: Type of message (e.g., text, image).
- status: Status of the message (e.g., read, unread).
- created_at: Timestamp for message creation.
- updated_at: Timestamp for last update.

2. Relationships:

- One-to-Many relationship between users and jobs (users.id -> jobs.user_id).
- One-to-One relationship between users and profile_details (users.id -> profile_details.user_id2).
- One-to-Many relationship between users and chatify for both user_id and friend_id.

2.9 GRAPHICAL USER INTERFACES

Home page



WELCOME TO CARESEEKER

Discover Opportunities for Care Seekers and Support Workers

Care and support workers are ready to help you with

Meet our care and support workers

Ahmed Ali
Karachi, Pakistan
Hi, I'm a compassionate support worker based in Karachi. With a focus on personal care, I strive to create a comfortable environment for my clients, promoting their well-being and independence.

Zainab Malik
Islamabad, Pakistan
I'm a dedicated support worker in Islamabad. My expertise lies in elderly care and companionship. Clients appreciate my empathy, reliability, and commitment to fostering a positive atmosphere.

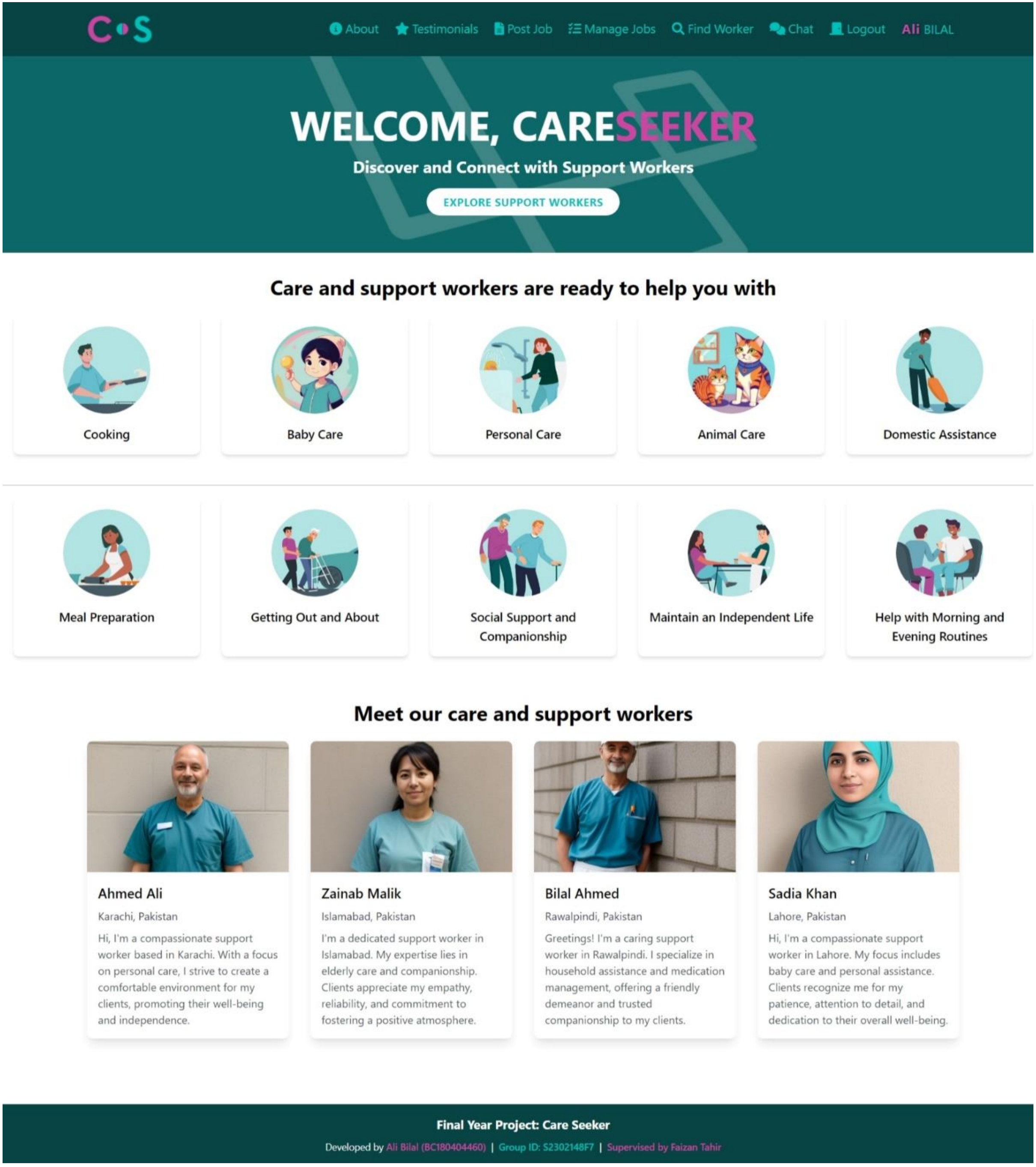
Bilal Ahmed
Rawalpindi, Pakistan
Greetings! I'm a caring support worker in Rawalpindi. I specialize in household assistance and medication management, offering a friendly demeanor and trusted companionship to my clients.

Sadia Khan
Lahore, Pakistan
Hi, I'm a compassionate support worker in Lahore. My focus includes baby care and personal assistance. Clients recognize me for my patience, attention to detail, and dedication to their overall well-being.

Final Year Project: Care Seeker

Developed by Ali Bilal (BC180404460) | Group ID: S2302148F7 | Supervised by Faizan Tahir

Home page as Care Seeker



WELCOME, CARESEEKER

Discover and Connect with Support Workers

Care and support workers are ready to help you with

Cooking

Baby Care

Personal Care

Animal Care

Domestic Assistance

Meal Preparation

Getting Out and About

Social Support and Companionship

Maintain an Independent Life

Help with Morning and Evening Routines

Meet our care and support workers

Ahmed Ali

Karachi, Pakistan

Hi, I'm a compassionate support worker based in Karachi. With a focus on personal care, I strive to create a comfortable environment for my clients, promoting their well-being and independence.

Zainab Malik

Islamabad, Pakistan

I'm a dedicated support worker in Islamabad. My expertise lies in elderly care and companionship. Clients appreciate my empathy, reliability, and commitment to fostering a positive atmosphere.

Bilal Ahmed

Rawalpindi, Pakistan

Greetings! I'm a caring support worker in Rawalpindi. I specialize in household assistance and medication management, offering a friendly demeanor and trusted companionship to my clients.

Sadia Khan

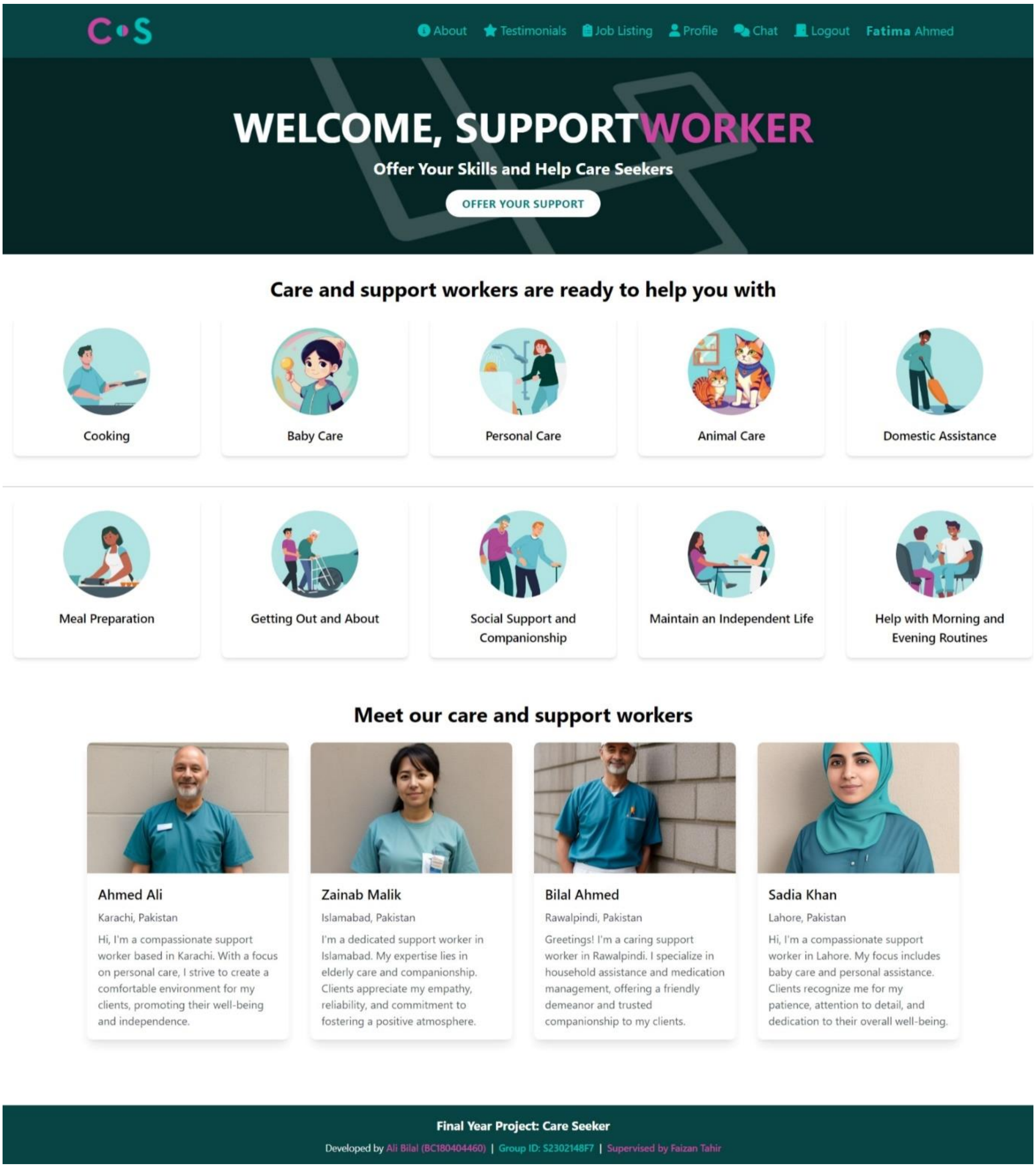
Lahore, Pakistan

Hi, I'm a compassionate support worker in Lahore. My focus includes baby care and personal assistance. Clients recognize me for my patience, attention to detail, and dedication to their overall well-being.

Final Year Project: Care Seeker

Developed by Ali Bilal (BC180404460) | Group ID: S2302148F7 | Supervised by Faizan Tahir

Home Page as Support Worker



WELCOME, SUPPORTWORKER

Offer Your Skills and Help Care Seekers

Care and support workers are ready to help you with

Cooking

Baby Care

Personal Care

Animal Care

Domestic Assistance

Meal Preparation

Getting Out and About

Social Support and Companionship

Maintain an Independent Life

Help with Morning and Evening Routines

Meet our care and support workers

Ahmed Ali
Karachi, Pakistan
Hi, I'm a compassionate support worker based in Karachi. With a focus on personal care, I strive to create a comfortable environment for my clients, promoting their well-being and independence.

Zainab Malik
Islamabad, Pakistan
I'm a dedicated support worker in Islamabad. My expertise lies in elderly care and companionship. Clients appreciate my empathy, reliability, and commitment to fostering a positive atmosphere.

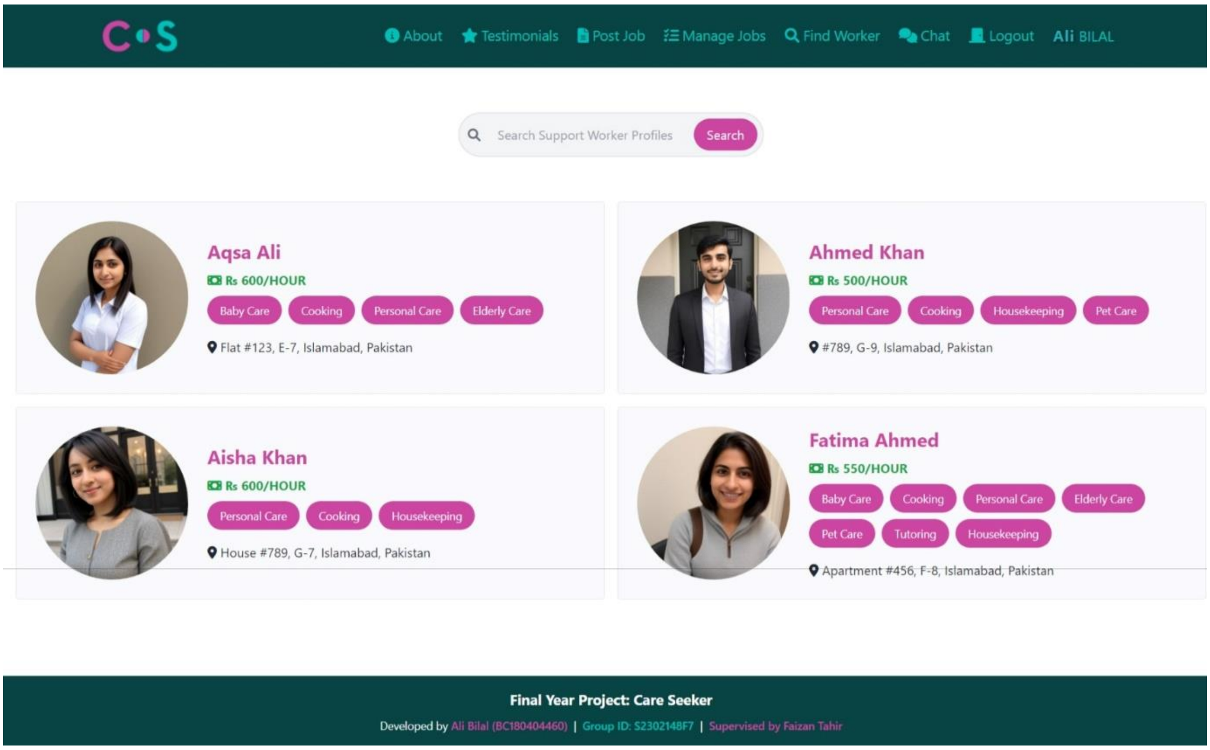
Bilal Ahmed
Rawalpindi, Pakistan
Greetings! I'm a caring support worker in Rawalpindi. I specialize in household assistance and medication management, offering a friendly demeanor and trusted companionship to my clients.

Sadia Khan
Lahore, Pakistan
Hi, I'm a compassionate support worker in Lahore. My focus includes baby care and personal assistance. Clients recognize me for my patience, attention to detail, and dedication to their overall well-being.

Final Year Project: Care Seeker

Developed by Ali Bilal (BC180404460) | Group ID: S2302148F7 | Supervised by Faizan Tahir

Worker List Page



Aqsa Ali
Rs 600/HOUR
Baby Care | Cooking | Personal Care | Elderly Care
Flat #123, E-7, Islamabad, Pakistan

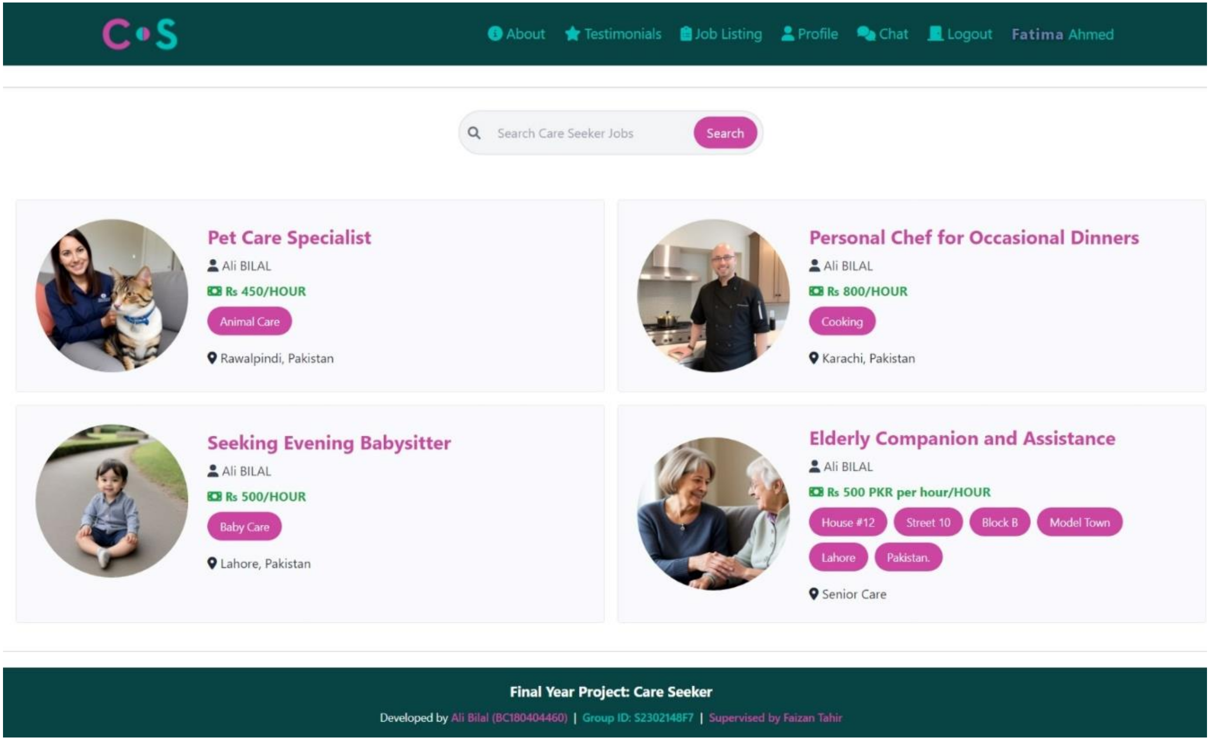
Ahmed Khan
Rs 500/HOUR
Personal Care | Cooking | Housekeeping | Pet Care
#789, G-9, Islamabad, Pakistan

Aisha Khan
Rs 600/HOUR
Personal Care | Cooking | Housekeeping
House #789, G-7, Islamabad, Pakistan

Fatima Ahmed
Rs 550/HOUR
Baby Care | Cooking | Personal Care | Elderly Care | Pet Care | Tutoring | Housekeeping
Apartment #456, F-8, Islamabad, Pakistan

Final Year Project: Care Seeker
Developed by Ali Bilal (BC180404460) | Group ID: S2302148F7 | Supervised by Faizan Tahir

Job List Page



Pet Care Specialist
Ali BILAL
Rs 450/HOUR
Animal Care
Rawalpindi, Pakistan

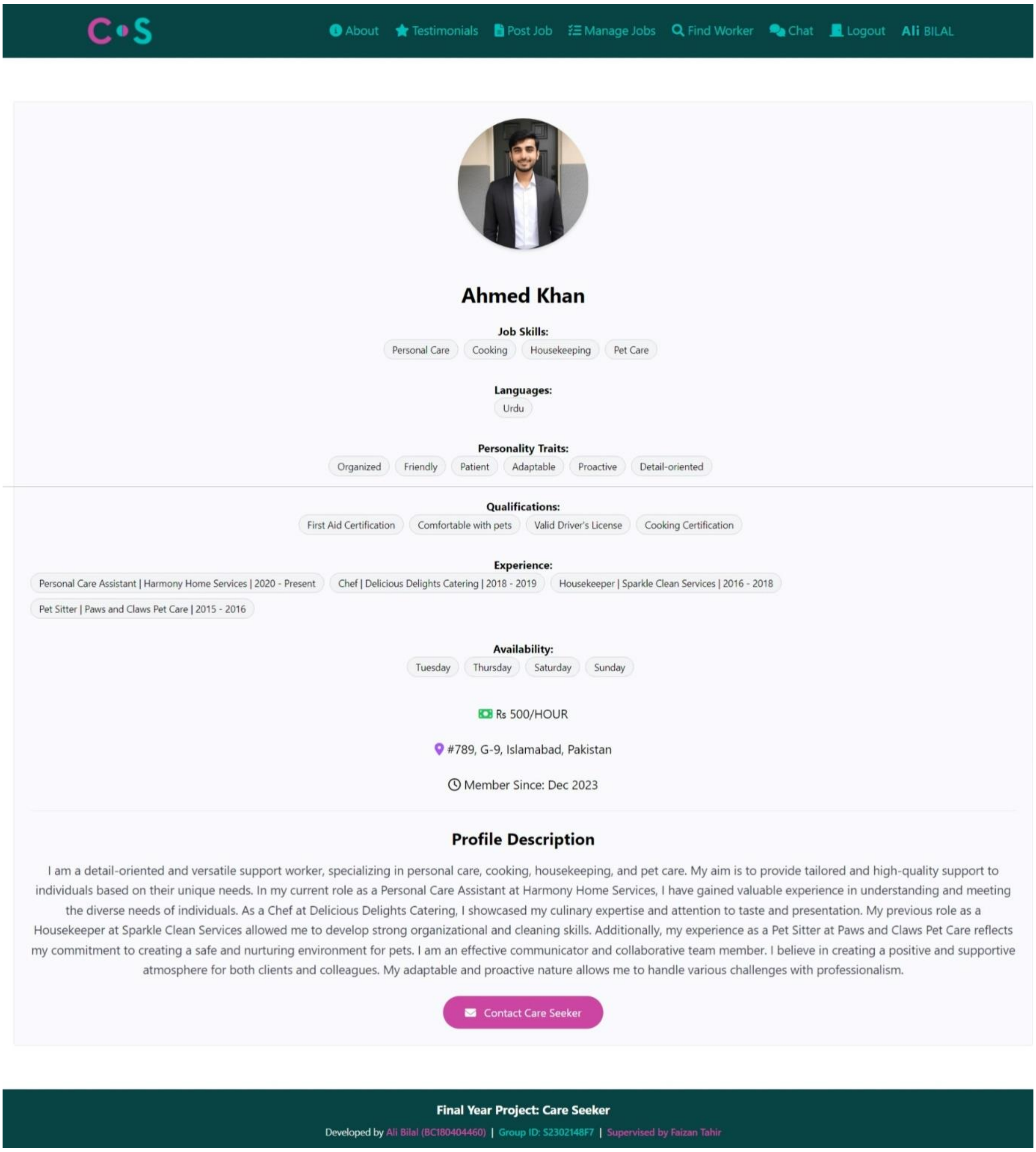
Personal Chef for Occasional Dinners
Ali BILAL
Rs 800/HOUR
Cooking
Karachi, Pakistan

Seeking Evening Babysitter
Ali BILAL
Rs 500/HOUR
Baby Care
Lahore, Pakistan

Elderly Companion and Assistance
Ali BILAL
Rs 500 PKR per hour/HOUR
House #12 | Street 10 | Block B | Model Town | Lahore | Pakistan.
Senior Care

Final Year Project: Care Seeker
Developed by Ali Bilal (BC180404460) | Group ID: S2302148F7 | Supervised by Faizan Tahir

Support Worker Profile



Ahmed Khan

Job Skills:
Personal Care | Cooking | Housekeeping | Pet Care

Languages:
Urdu

Personality Traits:
Organized | Friendly | Patient | Adaptable | Proactive | Detail-oriented

Qualifications:
First Aid Certification | Comfortable with pets | Valid Driver's License | Cooking Certification

Experience:
Personal Care Assistant | Harmony Home Services | 2020 - Present
Chef | Delicious Delights Catering | 2018 - 2019
Housekeeper | Sparkle Clean Services | 2016 - 2018
Pet Sitter | Paws and Claws Pet Care | 2015 - 2016

Availability:
Tuesday | Thursday | Saturday | Sunday

Rs 500/HOUR

#789, G-9, Islamabad, Pakistan

Member Since: Dec 2023

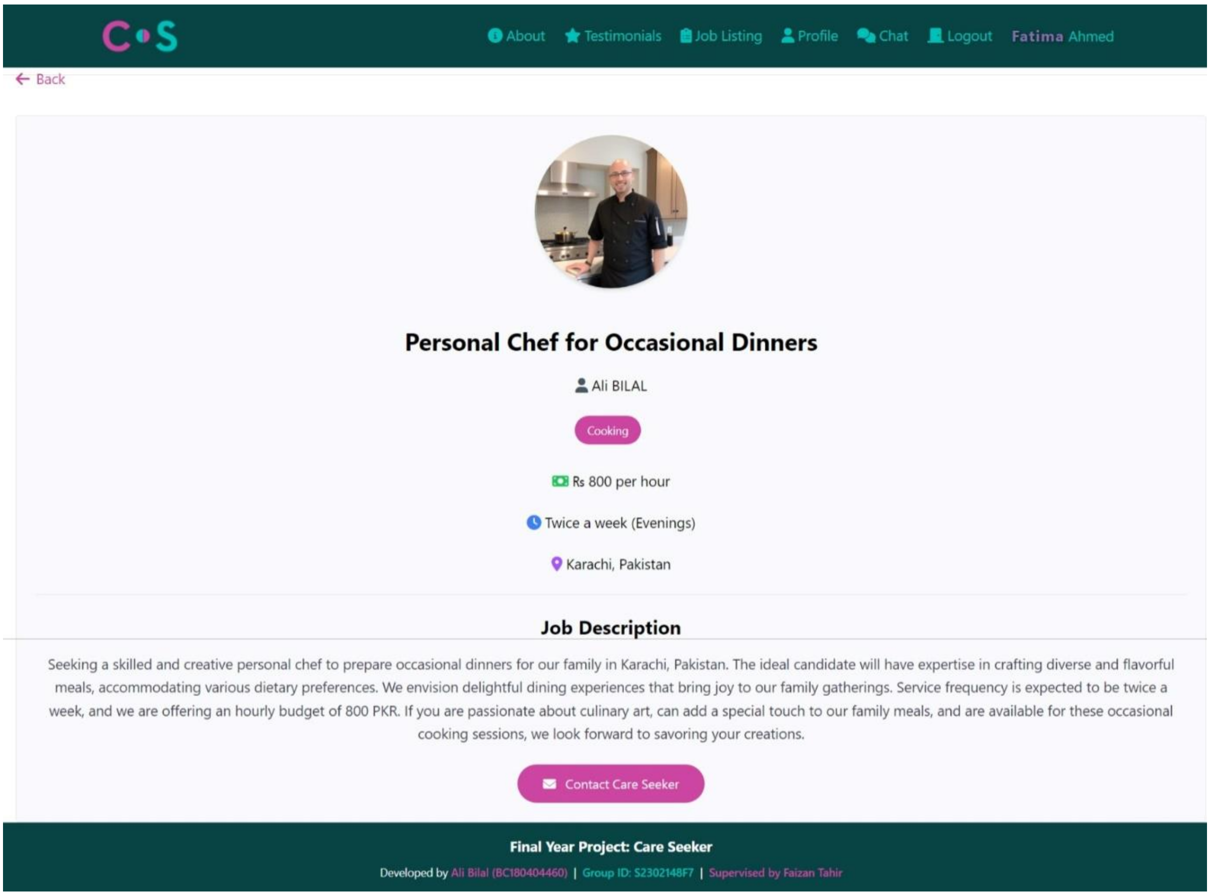
Profile Description

I am a detail-oriented and versatile support worker, specializing in personal care, cooking, housekeeping, and pet care. My aim is to provide tailored and high-quality support to individuals based on their unique needs. In my current role as a Personal Care Assistant at Harmony Home Services, I have gained valuable experience in understanding and meeting the diverse needs of individuals. As a Chef at Delicious Delights Catering, I showcased my culinary expertise and attention to taste and presentation. My previous role as a Housekeeper at Sparkle Clean Services allowed me to develop strong organizational and cleaning skills. Additionally, my experience as a Pet Sitter at Paws and Claws Pet Care reflects my commitment to creating a safe and nurturing environment for pets. I am an effective communicator and collaborative team member. I believe in creating a positive and supportive atmosphere for both clients and colleagues. My adaptable and proactive nature allows me to handle various challenges with professionalism.

Final Year Project: Care Seeker

Developed by Ali Bilal (BC180404460) | Group ID: S2302148F7 | Supervised by Faizan Tahir

Job Page



Personal Chef for Occasional Dinners

Ali BILAL

Cooking

Rs 800 per hour

Twice a week (Evenings)

Karachi, Pakistan

Job Description

Seeking a skilled and creative personal chef to prepare occasional dinners for our family in Karachi, Pakistan. The ideal candidate will have expertise in crafting diverse and flavorful meals, accommodating various dietary preferences. We envision delightful dining experiences that bring joy to our family gatherings. Service frequency is expected to be twice a week, and we are offering an hourly budget of 800 PKR. If you are passionate about culinary art, can add a special touch to our family meals, and are available for these occasional cooking sessions, we look forward to savoring your creations.

Final Year Project: Care Seeker

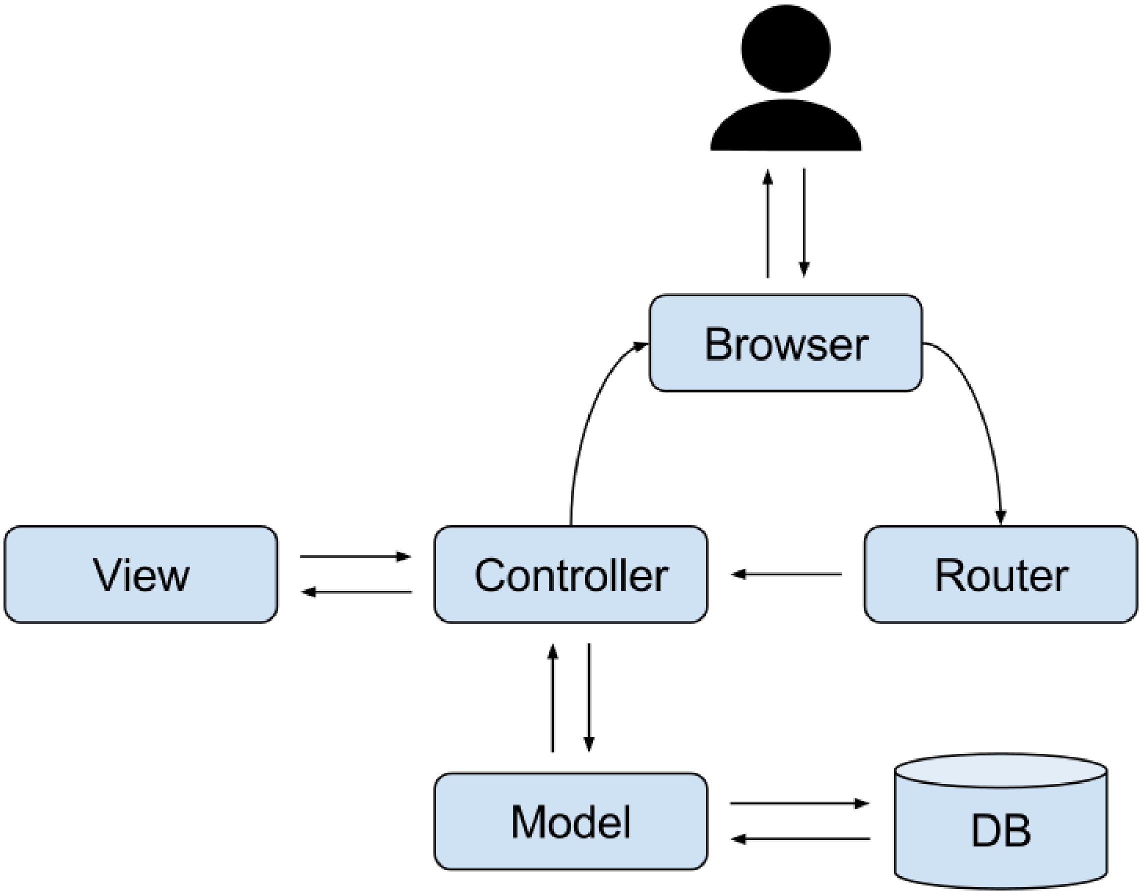
Developed by Ali Bilal (BC180404460) | Group ID: S2302148F7 | Supervised by Faizan Tahir

CHAPTER 3

DEVELOPMENT

DEVELOPMENT

3.1 DEVELOPMENT PLAN (ARCHITECTURE DIAGRAM)



MVC (Model-View-Controller) is a design pattern widely used in web development, and Laravel follows the MVC architectural pattern. Here's a brief overview of how MVC works in Laravel:

1. Model:

- The Model represents the data layer of the application.
- In Laravel, models are PHP classes that interact with the database. They encapsulate the business logic related to data manipulation.
- Models define the structure of database tables and establish relationships between them.

2. View:

- The View represents the presentation layer of the application.
- Views in Laravel are responsible for displaying the user interface and presenting data to users.

- Blade is the templating engine used in Laravel for creating views. It allows you to embed PHP code directly in the HTML.

3. Controller:

- The Controller acts as an intermediary between the Model and the View.
- Controllers handle user requests, process the input, interact with the model to retrieve data, and pass the data to the view for presentation.
- In Laravel, controllers are responsible for processing HTTP requests and returning an appropriate HTTP response.

Request Flow:
1. User Interaction:
A user interacts with the application by making an HTTP request (e.g., accessing a URL).

2. Routing:
Laravel's routing system determines which controller and method should handle the request based on the URL and HTTP method.

3. Controller Handling:
The specified controller method is invoked to process the request.
The controller may interact with the model to retrieve or manipulate data.

4. Model Interaction:
The model interacts with the database to perform CRUD (Create, Read, Update, Delete) operations.

5. Data Processing:
The controller processes the data obtained from the model and prepares it for presentation.

6. View Rendering:
The controller passes data to the view.
The view, typically created with Blade syntax, renders the HTML markup with the provided data.

7. HTTP Response:
The complete HTML response is sent back to the user's browser.

Advantages of MVC in Laravel:

- **Separation of Concerns:** MVC separates different aspects of the application, making it easier to manage and maintain code.
- **Modularity**: Each component (Model, View, Controller) is independent and can be developed, tested, and modified separately.

- **Reusability:** Code reusability is enhanced as models, views, and controllers can be reused across different parts of the application.
- **Scalability:** The modular structure allows for scalability by adding or modifying components without affecting the entire application.
- **Testability:** Each component can be tested independently, making it easier to write unit tests for controllers and models.

Laravel's implementation of MVC simplifies the development process by providing a structured and organized way to build web applications.

1. Initiation Phase:

- Understand the project's purpose and goals.
- Identify who is involved and what they will do.
- Gather initial information to kickstart the project.

2. Planning Phase:

- Define what the project needs and the features it will have.
- Document all the requirements in a clear way.
- Imagine how people will use the project through scenarios.
- Choose the approach for building the project.
- Create a visual timeline (Gantt chart) for project tasks.
- Select the tools and technologies to use, like Laravel, Tailwind CSS, Alpine.js, and Chatify.

3. Designing Phase:

- Plan how the project's parts will fit together (MVC, Tailwind CSS, Alpine.js).
- Create a diagram to visualize the project structure.
- Design the database structure for storing information.
- Craft the project's appearance using Blade templates and Tailwind CSS.
- Add interactive elements using Alpine.js.
- Plan how different sections of the project will work together seamlessly.

4. Implementation Phase:

- Set up the core framework of the project using Laravel.
- Build the different elements, focusing on data handling, user interface, and interactions.
- Use Tailwind CSS for a visually appealing design.
- Enhance interactivity with Alpine.js.
- Code the user registration and authentication processes.
- Develop profiles for support workers.
- Implement the job posting and application functionalities.
- Integrate the Chatify messaging system.

- Ensure the project works smoothly across various devices.

5. Testing and QA Phase:

- Conduct individual tests to ensure each part works correctly (Unit Testing).
- Verify the seamless integration of all components (Integration Testing).
- Test user interactions and project functionalities.
- Check the job posting and application procedures.
- Confirm the effectiveness of the messaging system.
- Ensure compatibility with different web browsers and devices.

6. Deployment Phase:

- Prepare the project for public use, including initial data setup.
- Deploy the project on a web server for online access.
- Monitor the project's performance and address any arising issues.

7. Maintenance Phase:

- Monitor user feedback and assess project performance.
- Address and fix reported issues promptly.
- Update documentation for clarity.
- Plan for future improvements and additional features.

REFERENCES

1. Laravel Documentation. (n.d.). "Laravel - The PHP Framework For Web Artisans." Retrieved from https://laravel.com/docs

2. Tailwind CSS Documentation. (n.d.). "Tailwind CSS - A utility-first CSS framework." Retrieved from https://tailwindcss.com/docs

3. Alpine.js Documentation. (n.d.). "Alpine.js - A rugged, minimal framework for composing JavaScript behavior in your markup." Retrieved from https://alpinejs.dev/start

4. Chatify Documentation. (n.d.). "Chatify - A Laravel Package for a Realtime Laravel Chat." Retrieved from https://christofferok.com/product/chatify

5. Laravel Course. “Traversy Media - Laravel From Scratch 2022 | 4+ Hour Course.” Retrieved from https://www.youtube.com/watch?v=MYyJ4PuL4pY

6. PHP Course. “Traversy Media - PHP For Beginners | 3+ Hour Crash Course.” Retrieved from https://www.youtube.com/watch?v=BUCiSSyIGGU

7. MySQL Course. “Traversy Media - MySQL Crash Course | Learn SQL” Retrieved from https://www.youtube.com/watch?v=9ylj9NR0Lcg

APPENDIX

A. Gathering & Analyzing Info:

A.1 User Surveys:

- Detailed summaries of surveys conducted to understand user needs and preferences.

A.2 Data Analysis Reports:

- Comprehensive reports presenting insights gained from the analysis of gathered information.

B. Designing The Project:

- **B.1 UI/UX Wireframes:**
  - Visual representations of preliminary design concepts for user interfaces and experiences.

- **B.2 Architecture Diagrams:**
  - Diagrams illustrating the project's architectural design for both frontend and backend.

- **B.3 Technology Stack:**
  - Detailed documentation on the chosen technologies and tools for project development.

C. Development:

- **C.1 Code Samples:**
  - Selected excerpts of key code segments showcasing the implementation of critical features.

- **C.2 Testing Documentation:**
  - Reports detailing the various testing phases, methodologies used, and the obtained results.

- **C.3 Deployment Guide:**
  - Step-by-step guide for deploying the developed project, including server and database setup.