D424 – Software Capstone

Task 2

Capstone Proposal Project Name: Caretaker Management Web Application (CMWA)

Student Name: WD

Table of Contents

Business Problem

The Customer

CareCircle Pvt Ltd (CC) is a growing caretaker placement agency operating in 5 cities across India. With a dedicated team of professionals, the company manages a sizable database of vetted caretakers catering to a diverse clientele with varying care needs. CC specializes in providing full-time 24/7 caretakers for a range of medical conditions, including dementia, cancer, stroke, post-hospitalization, and long-term chronic care. Their mission is to provide vetted, reliable, and compassionate full-time 24/7 caretakers to clients with various medical conditions, ensuring the highest quality of care and support.

The organization is led by a team of experienced professionals, including the CEO, operations managers, HR managers, and placement coordinators, who work closely together to ensure the efficient functioning of the business and the satisfaction of both caretakers and clients.

CC currently relies on traditional methods of managing client and caretaker information, including spreadsheets and manual record-keeping. As the company grows, there is a pressing need to adopt a more efficient and scalable IT solution to better manage the placement process.

Short-term Goals:

- Improve operational efficiency by streamlining the caretaker placement process.
- Increase the number of successful placements and reduce the turnaround time for clients.
- Enhance client satisfaction through better matching of caretakers to client needs.
- Develop a scalable IT infrastructure to support the company's growth.

Long-term Goals:

- Expand the company's presence to more cities across India, reaching a wider clientele.
- Continuously improve the quality-of-care services provided by maintaining rigorous vetting processes for caretakers.
- Establish CC as a leading and trusted caretaker placement agency in the country.
- Adapt to emerging healthcare trends and technological advancements to stay ahead of the competition.

CC aims to expand its operations to more cities in India, steadily increasing its market share and client base. The company's focus on providing high-quality care services and efficient placement processes is expected to result in sustained growth and a strong reputation in the industry.

Business Case

The completion of the Caretaker Management Web Application (CMWA) will directly benefit CC by addressing their operational challenges, streamlining processes, and providing a scalable platform for future growth. This will enable the company to achieve its short and long-term goals, ultimately leading to improved client satisfaction and a stronger market presence.

CC currently faces several challenges in managing their caretaker placement process. The use of traditional methods, such as spreadsheets and manual record-keeping, has led to inefficiencies and difficulties in tracking caretaker assignments, maintaining up-to-date caretaker profiles and client information, and quickly responding to changing circumstances. As a result, the company may experience delays in placements, reduced client satisfaction, and limited revenue growth.

The CMWA will address CC's operational challenges by providing a centralized and efficient platform for managing the caretaker placement process. The application will offer various functionalities that will significantly improve the company's operations:

- Centralized Database: The CMWA will consolidate all essential information related to caretakers and clients in a single platform, ensuring easy access and efficient management for administrators.
- Caretaker Profile Management: The application will allow administrators to create, update, and maintain comprehensive caretaker profiles, including personal details, qualifications, skill sets, and availability.
- Client Information Management: The CMWA will provide a secure platform for managing client information, such as contact details, service requirements, and location. This feature ensures that all necessary data is readily available to facilitate the caretaker placement process.
- Placement Reporting: The application will be equipped with a reporting module that provides information on the details of all caretaker placements at clients, and it can also identify unassigned or "on-bench" caretakers. This data will allow administrators to optimize placements, enhance revenue, and ensure that all candidates stay involved and efficient.

By implementing the CMWA, CC will significantly improve its operational efficiency, streamline the caretaker placement process, and enhance its responsiveness to client needs. This will result in faster placements, higher client satisfaction, and increased revenue growth, ultimately supporting the company's mission to provide high-quality, reliable, and compassionate care services to its clients.

Fulfillment

CMWA will be designed to fulfill the needs of CC by providing a comprehensive, user-friendly, and efficient platform for managing the caretaker placement process. Key aspects of the application include its appearance, basic functions, and outcome presentation.

The CMA will feature a clean and modern user interface, with intuitive navigation and clear labeling to ensure that users can quickly access and manage information. The application will be optimized for both desktop and mobile devices, allowing administrators to manage the placement process from any location and at any time.

- Dashboard: The CMWA will feature a central dashboard that provides an overview of the placement process, displaying key metrics such as the number of active caretakers and open client requests. This will enable users to monitor performance and identify potential issues or areas for improvement.

- Caretaker Management: Users will be able to create, update, and maintain caretaker profiles, including details such as qualifications, skills, and availability. The application will provide search and filter functionalities, allowing users to easily find and match suitable caretakers to client needs.
- Client Management: The CMWA will offer a secure module for managing client information, such as contact details, service requirements, and location. Users will be able to search for clients, view their details, and track ongoing placements.
- Placement Reporting: The application will also enable real-time tracking of unassigned or "on-bench" caretakers, ensuring maximum placement opportunities.
- Users will be able to access and perform various functions within the CMWA through role-based access control, ensuring that each user has appropriate permissions to perform their tasks. This will include functions such as creating and managing caretaker profiles, updating client information, monitoring placements, and generating reports.

By fulfilling these needs, the CMWA will empower CC to optimize its operations, enhance client satisfaction, and achieve its mission of providing high-quality, reliable, and compassionate care services.

SDLC Methodology

Given the nature of the CMWA and the need for an MVP (Minimum Viable Product) initially, the Agile Scrum methodology is a suitable choice for managing the project. Agile Scrum is an iterative, incremental approach to software development, which allows for flexibility, collaboration, and rapid adaptation to change. The Agile Scrum methodology is a good fit for this project because it:

- Allows for flexibility in the development process, enabling the team to quickly adapt to changes and incorporate feedback from the CC.
- Facilitates close collaboration between team members and CC, ensuring that the MVP meets their needs and expectations.
- Focuses on delivering a functional product in a short timeframe, which aligns with the 40-hour timeline for the MVP.

The Agile Scrum methodology consists of several key phases, which align with different parts of the project:

- Sprint Planning: During this phase, the team will create a backlog of prioritized features and user stories for the MVP. They will determine which features can be implemented within the 40-hour timeframe and plan the work accordingly.
- Sprint Execution: The team will work in short sprints, typically one week long, to develop the prioritized features. The team members will collaborate closely, implementing the features and conducting unit testing as they go.

- Daily Stand-ups: The team members will hold brief daily meetings to discuss their progress, and any challenges they are facing, and to coordinate their efforts. This will ensure that the project stays on track and any issues are quickly addressed.
- Sprint Review and Retrospective: At the end of each sprint, the team will hold a review to demonstrate the completed features to the customer and gather feedback. They will also conduct a retrospective to discuss what went well, what could be improved, and any adjustments to be made for the next sprint.
- Iterative Development: The team will continue to work in sprints, refining the MVP based on customer feedback and their own experience. This iterative approach ensures that the final product is tailored to the customer's needs and expectations.

Deliverables

Given the short 40-hour timeline for the MVP with two people working on it, the deliverables for each phase of the project will be tailored accordingly. Here's a revised list of deliverables:

- Project Scope Document: A brief document outlining the objectives, high-level requirements, and goals for the MVP. This document will serve as a reference for the team during the development phase.
- Wireframe: A low-fidelity, rough representation of the application.
- User Stories and Prioritized Feature List: A collection of user stories and a list of prioritized features for the MVP. This list will help the team focus on the most essential functionalities given the limited timeframe.
- Working Prototype/MVP: A functional prototype of the MVP, implementing the prioritized features from the planning and design phase. This prototype should be able to demonstrate the core functionality of the CMWA. Example: A working prototype with basic caretaker profile management and client information management features.
- Source Code: The complete source code for the CMWA MVP, following best practices and coding standards to ensure maintainability and scalability. The source code will be written in Node.js and stored in a version control system.
- Deployment Instructions: A concise guide on how to deploy the MVP.
- User Guide: A brief user guide to help CC administrators understand the basic features and functionality of the MVP, enabling them to use the application effectively.
- Test Cases and Results: A set of basic test cases for the implemented features, along with the test results to ensure that the MVP is functioning as expected.

Deployment Plan and Outcomes

The deployment plan for the MWA MVP, given the short 40-hour timeframe with two people working on it, will focus on a streamlined and efficient process to put the software into the production environment. The deployment plan will be executed in the following steps:

- Validation and Verification: This process will take place concurrently during the development phase. As the team implements features, they will perform unit testing to ensure the functionality works as intended. After the MVP is completed, a brief integration testing will be conducted to verify that all components work together seamlessly. The validation and verification process will be managed by both team members who have been working on the project.
- Preparation and Coordination: The two team members will collaborate with the customer, CC, to determine the appropriate time for deployment, ensuring minimal disruption to their operations. The deployment plan will be shared with the customer, outlining the steps to be followed and any necessary coordination required.
- Deployment to Production Environment: The implementation of the MVP is relatively simple, given its minimal scope. One team member will act as the Project Manager, overseeing the deployment process and coordinating with any necessary personnel, such as web administrators or IT support staff from CC The other team member will be responsible for the technical aspects of deployment, ensuring that the application is properly configured and set up on the chosen DigitalOcean cloud platform. They will also ensure that the online Cockroach PostgreSQL database is correctly set up and connected to the application.
- User Training and Support: The team will provide a brief user guide for CC administrators to help them understand and navigate the MVP. As part of the deployment process, the team will schedule a short training session (via video conference or in-person, depending on the feasibility) to walk the administrators through the basic features and functionality of the application. This training session will be conducted by one of the team members who has worked on the project.
- Post-deployment Monitoring and Support: After the MVP is deployed and in use, the team will remain available to address any issues or concerns that may arise during the initial days of operation. They will provide prompt support and bug fixes as needed, ensuring a smooth transition for CC.

Project Timeline

For this section, you'll need to look at the phases of the project and provide information about the time required to complete each phase.

Phase	Milestone/Task	Deliverable	Description	Dates
Planning & Design	Task 1	Project Scope Document & Wireframe	Meeting with customer and procedure review	3/27/2023 – 3/28/2023
Planning & Design	Task2	User Stories/Prioritized Feature List	A collection of user stories and a list of prioritized features for the MVP	3/27/2023 – 3/28/2023
Development	Task3	Working prototype	A working prototype/MVP of the application	3/28/2023 – 3/30/2023
Development	Task4	Source code	The completed source code for the MVP storied in a versioning system such as Git	3/28/2023 – 3/30/2023
Documentation	Task5	Deployment Instructions	Deployment guide	3/31/2023 – 4/02/2023
Documentation	Task6	User Guide	A user guide detailing steps to create/manage caretaker profiles and client information using the MVP	3/31/2023 – 4/02/2023
Documentation	Task7	Test Cases & Results	A document containing test cases and results for caretaker profile creation, updating, and deletion.	3/31/2023 – 4/02/2023

Environments and Costs

Programming Environment

To complete the CMWA MVP project using Node.js with Pug, Tailwind CSS, and CockroachDB, the following hardware and software requirements are necessary:

- Hardware Requirements:
    - Development Machines: Each team member will require a computer with sufficient processing power, memory, and storage to handle the development tasks. The machines should have a stable internet connection for accessing the online database and other cloud-based services.
- Software Requirements:
    - Node.js: The project will utilize Node.js as the server-side runtime environment. The team members should have the latest stable version of Node.js installed on their development machines.
    - Pug: Pug will be used as the template engine for the application, allowing the team to build reusable HTML templates with minimal code duplication. The team members should install the necessary Pug packages and plugins as required.
    - Tailwind CSS: Tailwind CSS will be used for styling the application's user interface. The team members should have the latest version of Tailwind CSS installed and configured for their development environment.
    - Code Editor: A suitable code editor or integrated development environment (IDE) is required for writing and managing the application's code. Some popular choices include Visual Studio Code, Atom, or Sublime Text.
    - Version Control System: A version control system, such as Git, will be used to manage and track changes to the application's source code. The team members should have Git installed on their development machines and be familiar with its usage.
    - CockroachDB: The project will utilize CockroachDB as the online database solution. The team members will need to create an account and set up a CockroachDB instance on a suitable cloud platform. They should also install the necessary database drivers and packages for Node.js to interact with CockroachDB.
    - Deployment Platform: The application will be deployed on a DigitalOcean cloud platform. The team members should create accounts on the chosen platform and familiarize themselves with the deployment process.
    - Browser: A modern web browser, such as Google Chrome or Mozilla Firefox, is necessary for testing and debugging the application during development.

Environment Costs

For the CMWA MVP, the costs associated with the software application can be categorized into initial setup costs and recurring costs.

- Initial Setup Costs: Domain registration is $20.
- Recurring Costs:
    - Cloud Database Maintenance: The Cockroach DB instance will have an annual fee, which covers database maintenance, storage size, and guaranteed uptime. The exact cost depends on the chosen plan and provider, but an estimate could be around $1200 per year.
    - DigitalOcean cloud platform: The web server, which hosts the application, will have an annual fee of approximately $1300. This fee includes their standard support.
    - Docker: This will hold the private images and will have an annual fee of $120 for a single user.

Overall, the online infrastructure costs for the CMWA MVP are relatively minimal, with some initial setup costs and recurring fees for the database maintenance and deployment platform.

Human Resource Requirements

To complete the CMWA MVP with two people working for 40 hours each, the labor cost can be calculated as follows:

- Labor cost per person = Hourly rate per person * Total hours worked
- Labor cost per person = $100/hour * 40 hours
- Labor cost per person = $4,000

Since there are two people working on the project, the total labor cost can be determined by multiplying the labor cost per person by the number of people:

- Total labor cost = Labor cost per person * Number of people
- Total labor cost = $4,000 * 2
- Total labor cost = $8,000

Hence, the time required to complete the application is 40 hours per person, and the total labor cost for the project is $8,000.

Validation and Verification

Given the short 40-hour timeframe for the MVP, the validation and verification process will be focused on ensuring that the software application functions sufficiently well to meet CC's needs. The testing methods will be as follows:

- Unit Testing: As the team members implement features during the development phase, they will perform unit testing to validate that each component works as intended. This will be done by writing test cases for individual modules and executing them to ensure the expected output is achieved. Both team members will be responsible for conducting unit testing for the code they write.
- Integration Testing: Once the MVP is complete, a brief integration testing will be conducted to ensure that all components work together seamlessly. This will involve testing the interaction between various modules, such as the caretaker profile management and client information management modules. The integration testing will be performed by both team members, leveraging their knowledge of the implemented features.
- User Acceptance Testing (UAT): CC will perform acceptance testing prior to taking ownership of the application. This will involve multiple testing sessions with multiple users from the customer's side to ensure the application meets their requirements and expectations. The team will provide necessary support and guidance during the UAT process, addressing any issues that may arise.
- Functional Testing: Functional testing will be carried out to validate that the application provides all the required functionality specified in the project scope. This will include testing various use case scenarios to ensure the application behaves as expected under different conditions. The functional testing will be conducted by both team members, with inputs from the CC during the UAT phase.
- Bug Fixing and Re-testing: Any issues or bugs discovered during the testing phases will be promptly addressed by the team members. Once a bug is fixed, the related test cases will be re-executed to ensure the issue has been resolved and the application functions as expected.

By employing these testing methods, the CMWA MVP can be validated and verified to meet CC's needs, ensuring a successful and functional product.