|  |
| --- |
| ICT30017 - ICT PROJECT A |
| PROJECT SPECIFICATION |
| PORTFOLIO TASK 3 |

|  |
| --- |
| SUBMITTED: 08/0/2024.  SUBMITTED: 22/03/2024 |

GROUP 2

**Table of Contents**

1. [Acknowledgement of Country 2](#_Toc141109977)
2. [Contributions 2](#_Toc141109977)

2.1. [Rubie Stannard 2](#_Toc141109978)

2.2. [Qiao Jun Chan 2](#_Toc141109979)

2.3. [Dylan Morrison 2](#_Toc141109978)

2.4. [Mingyuan Wang 2](#_Toc141109979)

2.5. [Henry Hua Rong Wang Hong 2](#_Toc141109979)

2.6. [Tan Dat Do 2](#_Toc141109979)

1. [Solution Approach 3](#_Toc141109977)
2. [Challenges 7](#_Toc141109977)

4.1. [Technical 7](#_Toc141109978)

4.2. [Process Related 7](#_Toc141109979)

1. [Incremental Development Plan 9](#_Toc141109977)
2. [Tracking and Monitoring 11](#_Toc141109977)
3. [Quality Plan 12](#_Toc141109977)
4. **Acknowledgement of Country**

We acknowledge the traditional custodians of the land on which we gather, the Wurundjeri people of the Kulin Nation, and pay our respects to their elder’s past, present, and emerging. We recognize their continuing connection to the land and culture and honour the significant role they play in shaping the identity and history of this region.

1. **Contributions**

**2.1. Rubie Stannard: 103982732**

I contributed to this task by making the document, adding the cover page, table of contents, tables, headings, writing the acknowledgement of country, and completing the incremental development plan.

**2.2. Qiao Jun Chan: 103490604**

I contributed to this task by seeking solution approach based on the problems have been defined. Then, do research on the alternate approach.

**2.3. Dylan Morrison: 101111673**

I contributed to this task by focusing on the Quality Plan, and engaging in discussion around the other aspects of this project plan.

**2.4. Mingyuan Wang: 104195667**

I contributed to this documentation by pointing out some challenges.

**2.5. Henry Hua Rong Wang Hong: 104792738**

I participated in the team discussion and expressed my own opinion. I preliminarily completed the Tracking and Monitoring part.

**2.6. Tan Dat Do: 103498255**

I contributed to this portfolio task 3 by finish quality plan and backup with other members task.

1. **Solution Approach**

In the realm of aged care services, the prevalence of disparate and disconnected software systems represents a significant challenge. This fragmentation leads to inefficiencies, increased risks, and compromises in patient care. Our project aims to revolutionize this landscape by integrating these systems into a cohesive software solution, thereby mitigating risks and enhancing care quality.

The current landscape is characterized by the use of multiple, unconnected software solutions, each catering to different facets of aged care. This disconnection severely hampers effective communication and coordination among caregivers, adversely affecting the timeliness and accuracy of care delivery. Such inefficiencies can lead to missed tasks, overlooked health concerns, and delayed responses, ultimately compromising patient outcomes. Furthermore, inconsistencies in critical data such as medication dosages and inventory management can lead to grave errors in patient care. Our project seeks to address these issues by creating a unified software environment, significantly improving care quality and operational efficiency. The project will unfold in several strategic phases:

**Phase 1: Analysis**

In the analysis phase, we aim to pinpoint specific inefficiencies and risks within the current software landscape. This will be achieved through consultations with key stakeholders, including caregivers, IT staff, users, and management. These discussions will be facilitated using questionnaires, short answers, and rating forms. The primary goal is to assess existing software solutions comprehensively, gaining insight into their functionalities, limitations, and usage patterns.

* Demographic questions: Age, Height, Gender, Health history, Blood types, Ethnicity, etc
* Identify target user (Health Assessment): Glucose level, Smoking status, Family health history, Past surgery, Current meditation, Allergy history, Living habits
* Features expectation: Remainder, Heart tracking, Appointment, Contact storage

**Phase 2: Solution Design**

By observing the performance of existing software solutions, we aim to understand their functionalities, limitations, and their application in daily operations. The objective is to ensure that the new system promotes seamless communication and data sharing among caregivers, supporting real-time updates and alerts. Our plan includes developing a web-based management portal that consolidates all essential functionalities into a single platform. This unified strategy is designed to overcome the current fragmentation, thereby enhancing efficiency, communication, and care coordination.

1. Responsive Web Design

Ensure the website is fully functional and user-friendly across various devices and browsers through responsive design principles. The use of HTML, CSS, and javascript for front-end development. For the homepage design, an intuitive navigation menu easier users accessible across pages, consider a fixed navigation bar that remains visible when users scroll. Ensure text contrasts sufficiently against the background, links are distinguishable and buttons are easily clickable. Highlighting key sections such as “About Us”, “Services”, “Patient Portal”, “Contact”, and “Support”. An appropriate color scheme like green will evokes thrust and calmness, catering to the target audience of aged care providers and families (How to Choose the Perfect Color Scheme, 2023).

In the login page, a secure login system is essential with encrypted password. Adjust the login page to prompt for a username and password initially. After successfully entering these, the user should be directed to a second page or modal window asking for the second factor. Create a setup page where users can enable 2FA, choose their preferred method like SMS, email, authenticator app, and complete the necessary steps to link their account with the chosen method. During the initial authentication, implement the first authentication step to verify the username and password against users’ database securely. However, in the stage of 2FA method integration, use an API service like Twilio or sendgrid send a code to the user’s phone or email.

* Twilio provides a broad range of communication apis (Application Programming Interfaces) that allow software developers to programmatically make and receive phone calls, send and receive text messages, and perform other communication functions.
* Sendgrid which was acquired by Twilio, specializes in email services. It provides a cloud-based service to send transactional and marketing emails.

1. Database Integration

Utilize a relational SQL database management system for secure and efficient data storage and retrieval, ensuring seamless integration with the web-based management portal. SQL (Structured Query Language) provides a flexible and efficient means of storing, retrieving, and manipulating these data. In the initial database scheme design we will create an individual table for differentiate the information respectively are patients table, caregivers table, family emergency contact table, care plans table, medication managemen table, incident table, financial information table, family and social history table, staff scheduling table, dietary preferences table, and appointment logs.

To establish a connection to a database and manage data, PHP is widely used in conjunction with SQL (Structured Query Language), the standard language for managing and manipulating relational databases. The integration of PHP with SQL databases allows for the development of applications that can dynamically generate web pages. This will be shown while we start developing the software.

1. Feature-Specific Pages

Develop separate HTML templates for each feature.

* Staff details page: List of all staff members with details like name, role, qualifications, contact information, and work schedule. Features like add, edit, delete staff records, search and filter capabilities by name, role, or qualification, view detailed profiles for each staff member.
* Patient information page: Personal details (name, DOB, address, phone number, email address), medical history, care plans, medication schedules, and family emergency contacts. Features like add, edit, and delete information in each sub-section, access and update health history details, including conditions, treatments and allergies, manage care plans, including activities, goals and progress notes, view and update emergency contact details.
* Appointment page: Upcoming and past appointments for each patient, including date, time, healthcare provider, and purpose. Features like schedule new appointments, with the ability to specify the reason and select the provider, edit and cancel existing appointments, and reminder settings for upcoming appointments.
* Staff scheduling page: Schedules of all staff members, showing shifts, location (whether in residency or in aged care centre), and special assignments. Feayures like create and manage shifts for staff, including start and end times, and assign staff to shifts, view schedules by day, week, or month, and swap shifts between staff members with proper authorization.
* Payment page: Billing details for services provided to elderly individuals, including itemized services, total amount, and payment status. Features like generate new invoices for payment, with the ability to add services and calculate totals automatically, mark invoices as paid and manage payment methods, access and download past invoices for record-keeping.
* Facility management page: : Information on facility resources, including room assignments, equipment, and inventory levels of medical supplies and commodities. Features like manage room assignments and availability for patients, track equipment status, including maintenance schedules and service history, inventory management for medical supplies, including stock levels, reorder alerts, and order placement.

1. Real-Time Communication

Integrate a real-time messaging function for effective communication between carers and families, enhancing the support network for elderly individuals. Design a clean and intuitive chat interface within the aged care system. This should include a message input field, conversation history, and indicators for message status (sent, delivered, read).

The websocket protocol is a communications protocol that provides full-duplex communication channels over a single TCP (Transmission Control Protocol) connection. With this approach, a server can send data to a browser without the browser explicitly requesting it, making it a great choice for real-time applications. Firstly, begins with an HTTP (Hypertext Transfer Protocol) request from the client (usually a web browser) to the server, with an "Upgrade" header indicating the request to upgrade the connection to websocket. Secondly, If the server supports websocket, it responds with an HTTP 101 status code (Switching Protocols), agreeing to switch protocols from HTTP to websocket. Once the handshake is complete, the protocol allows for data to be sent back and forth between the client and server in real-time. This data is sent in "frames", and both text and binary data can be transmitted (What is web socket and how it is different from the HTTP, 2023).

**Phase 3: Implementation and Evaluation**

Engage with end users throughout the development process to gather feedback and make necessary adjustments, ensuring the solution aligns with user needs and expectations. Aged care software often requires customization to meet the specific needs of the facility or organization. To ensure a seamless flow of information, the new software must be integrated with existing systems such as electronic health records, billing software, and other critical operational tools. After that, training for staff and caregivers is vital to ensure they are comfortable and proficient with new software will include hands-on training sessions, the creation of user manuals and guides, and the establishment of a support system to address any questions or issues that arise during and after the implementation. Before going to publish, we will conduct unit testing, system testing, and user acceptance testing with a select group of end-users.

**Phase 4: Monitoring and Feedback**

It is an ongoing process for ensuring that software continuously meets the evolving needs of aged care providers, caregivers, and residents. Continues user feedback collection by setting up channels for collecting feedback from all user groups which including caregivers, administrative staff, IT personnel, and potentially residents or their families. Based on the insights gained from monitoring and feedback, develop and implement updates to address identified issues, add new features, and enhance overall performance. This may involve minor adjustments or more significant enhancements based on evolving user needs and technological advancements.

**Alternate Approaches**

Considering alternative approaches can provide flexibility and possibly uncover efficiencies or technologies better suited to specific aspects of the system.

* Utilizing AI and Machine Learning: Incorporate AI and machine learning algorithms to analyze patient data, predict health trends, and provide insights for personalized care plans. This can include predictive analytics for patient health outcomes, automated alerts for abnormal health indicators, and AI-driven suggestions for care plan adjustments.
* Blockchain for Data Security and Sharing: For secure and immutable record-keeping, especially for patient health records and transactions, blockchain technology can be an innovative alternative. It can enhance data security, patient privacy, and facilitate secure sharing of information between authorized parties.
* Mongodb: Ideal for applications dealing with large volumes of data and where the data structure can change over time, such as big data applications, content management systems, and mobile apps with rapidly evolving features.
* Django for python: Allows developers to interact with databases using Python code instead of SQL, making database operations easier and more secure.

1. **Challenges**

When working in a team to develop software, encountering technical and process related challenges is inevitable. This section delves into these potential challenges that may arise during the development of the website. By identifying and addressing these challenges proactively, we aim to mitigate risks, optimize project outcomes, and ensure the successful delivery of a robust and reliable software solution.

**4.1. Technical**

**System integration:** Integrating various modules and functionalities may require dealing with interfaces and data exchange between different systems, ensuring seamless collaboration among them. For example, an elderly care management system consists of a module responsible for staff scheduling and for managing family contacts, during system integration, it's essential to ensure that these two modules can communicate with each other and share data. Therefore, we need to communicate and share the data in time by discord.

**Technology stack selection:** Choosing the appropriate technology stack to support system development and operation is a challenge, considering factors such as system scalability, performance requirements, extensibility, and our knowledge restriction. We studied some PHP and simple database knowledge, however, considering some complex operation needs frames to solve, we can make the system scalability short and keep the system as simple as possible to finish development based on what we have learned.

**User experience design:** Designing intuitive interfaces for elderly people and other users can be challenging, as elderly individuals may not be familiar with technology, necessitating simplifying complex operations and providing easily understandable interfaces. Therefore, we will use Figma to interact with each other and get feedback in time.

**4.2. Process Related**

**Lack of communication:** A lack of communication among team members will lead to misunderstandings. To prevent this from occurring we will have weekly team meetings and communicate daily.

**Poor coordination:** Having no team coordination will result in duplication errors and inefficiencies. In order to make sure this doesn’t happen we will use Jira and GitHub to ensure team members are uploading their work and updating other members on the work progress.

**Unclear expectations:** Expectations that are ambiguous or frequently change will lead to confusion and misalignment among team members, damaging the project. Regularly reviewing the progress and making sure everyone is on the same page in terms of expectations will mitigate this.

**Time limit:** For a project, 6 to 8 weeks may not be sufficient for thorough development and testing. To address this, we must focus on the important functionalities and iteratively deliver increments to achieve the goals in the timeline.

**Project management and team collaboration:** Effective project management and team collaboration mechanisms are essential for developing complex software systems to ensure timely delivery and maintain effective communication and collaboration among team members.

1. **Incremental Development Plan**

This project will be done in two sprints, with the first sprint being the setup and development stage in which all of the code will be done, and the second sprint being for finalisation where all of the testing and necessary changes will be made. Both sprints are separated into three weeks, with week one of the overall sprint starting in week five of the unit.

In the first sprint we will focus on creating a prototype of the website so we have a frame to work with so we know why we’re coding and what the code we’re developing will look like once it’s all finished. We will also focus on setting all of the necessary pages up so everyone has the same stuff to work on. The final thing we’ll focus on in the first sprint is coding. As we go along the first sprint, after each task is completed, we will evaluate it and iterate it if necessary. We will use this sprint to get all of the coding finished so we can use the second sprint to make the necessary changes.

This leads into the second sprint where testing for errors and bugs will be done, and feedback will be given to us about the function and design of our website. After this has been acquired, we will make changes according to the feedback, and then test the code again for errors. Lastly, we will go over the website as a team and decide if we’re all happy about the outcome, or if small changes need to be made. This shouldn’t need to be done as we’ll be iterating during the first sprint.

A close-up of a computer screen

Description automatically generated

A diagram of the sprint plan is seen above. As mentioned earlier, the first sprint is where all of the setting up and the main development will happen. A detailed list of tasks for sprint one is below, with the list starting at the first week where the setup will happen. The following tasks will be competed in the first week:

1. Design a website prototype:

* Before we can start the development, we need to have an idea as to how we want our website to look. This can be achieved by creating a prototype using online resources, or by drawing, which will be done by Rubie. After the prototype is made the team can discuss what to change.

1. Make HTML, CSS, PHP, and JavaScript pages:

* Once a prototype has been decided on, the pages for the code must be made. This includes pages for the front-end stuff like the HTML matching CSS code, and pages for the back-end PHP and JavaScript code. Rubie will be responsible for creating the pages, which will ensure that no one creates things that don’t need to be created.

The second week will be where the development starts, with the focus being on the front-end design. In order to complete this in a week, the tasks to be done are:

1. Develop the HTML code:

* Once the pages have been created, the next step is to populate them with the HTML code. The HTML code is part of the front-end development and is necessary for the website to function. Although Maxy is the front-end developer, and Tan is the backup developer, Tan will be responsible for coding the HTML, so everyone contributes equally.

1. Develop the CSS code:

* Also needed for a functioning, and looks, is another front-end issue, which is the CSS. The CSS must be coded to suit the HTML. Coordinating and ensuring each page matches will take some time, and this will be done by Henry.

The third and final week of the first sprint will be the back-end design. This is where all of the code has to be completed, until we test the website and get feedback and make the necessary changes. The last week requires the following tasks to be done:

1. Develop the PHP code:

* This project is all about information, and the way to keep information on a website is to create a database through PHP. Creating the databases will be Maxy’s responsibility as he is the back-end developer and the only one who wrote PHP as their skill.

1. Develop the JavaScript code:

* Still to be decided is whether or not JavaScript will be used, but if it is it will be figured out in the prototyping stage. In the case of JavaScript being needed, Dylan will be responsible for coding this.

1. Create data for the databases:

* This website will have sections for data to be entered, but we can’t know if this will work unless we have data of our own to use. Making the data will be Qiao’s job. She can do this on a document for someone else to enter, or she can enter it herself and let the team know how it functions.

Assigning one week to make the pages, another week for the front-end development, and the final week of sprint one being for the back-end development ensures that no crossover with the code and GitHub repository will happen. Separating the tasks into specific stages will mean that every member sticks to their role and does an equal amount of work throughout the project.

1. **Tracking and Monitoring**

The tool our team will use to track and monitor project progress in the first sprint is Jira. We will use a Gantt chart to create a clear list of tasks for the first sprint, track the progress of each task, and assign them to the corresponding member for them to complete in the timeframe. After completing a task, team members need to update Jira to ensure that other members know that the task has been completed. Through the flexibility of using a Gantt chart, task allocation can be adjusted in time if things are not completed in a timely manner.

In addition to using Jira, the team will hold a weekly meeting where members will report the current work results, follow-up with plans, and discuss problems that are encountered to get help from other members. The meeting will ensure that each team member is aware of the progress and problems of the entire team and can communicate and coordinate solutions to problems in a timely manner.

Another measure we’ll use to track the progress is GitHub. If members are completing their allocated tasks, and marking it as complete on the Gantt chart, they should upload their work to the GitHub repository for other members to download. If this isn’t the case, members will know that a task hasn’t been completed.

1. **Quality Plan**

As we start developing the system, we must ensure the quality of our project in order to deliver an acceptable solution that meets the project requirements. The quality plan outlines our target quality expectations, strategies for assuring quality, and planned testing activities for the first sprint.

**Target quality:**

* All features should be intuitive to the user, providing clear feedback where suitable in the case of user error.
* Web pages should be organised in a logical and clear manner, to make all features easily accessible and promote usability.
* Colour blindness should be taken into consideration when choosing the colour scheme for all pages.
* Users can only access the specific features and information relevant to them, ensuring client confidentiality.

**Steps to assure quality:**

* Ensure that incorrect user inputs provide concise instruction on the error made, so the user can correct the mistake. If the mistake is commonly made, reassess implementation to avoid it as the instruction is not intuitive enough.
* If certain pages/features are not found quickly without prior instruction on where to find them, reconsider the topology of the website based on the attempts of the users. If they are not able to locate the page without instruction, use the actions they take to guide an alternative topology for the web pages.
* Colour blindness can be overcome through focus on methods such as contrast, and is important to consider to ensure the portal can be used by all who need it.
* Authenticate the user on every page, to prevent malicious access to features/data that the user should not have access to.

**Planned testing (complete with a test user who is provided a persona and relevant instructions):**

* Intentionally fill forms with incorrect/irrelevant data and assess the test user to see how they respond to the error messages provided. Not all data will be incorrect, and their task could be continued to find the errors in prior instruction and overcome them (i.e., data in wrong spots).
* Task the test user to find certain features, or interact with them, timing them and recording their process of looking for said features.
* Colour blindness testing can be outsourced through test users if they experience it, else can be done through online applications to provide feedback on colour combinations and their visibility to the varying types of colour blindness.
* A project member could sign in with varying levels of access, testing each on its ability to access every feature to confirm that the authentication requirements align with the access of each web page (i.e., an aged carer staff member attempts to access the billing information that should be restricted to administrative staff only).