Developing a Testing Strategy

Image Generated by Dalle-E 2 from chat gpt

Prepared for

Khalis Ahmed

Prepared by

Jean-Pierre Nde-Forgwang
Jonah Scott
Nicholas Gonzalez
Valentine Jingwa
Group 7

Security System, Cohort E

School of Advance Digital Technology

SAIT

31 January 2024

Vita

"Team constitution"

Goals: We hope to achieve at least a 95% for our project and this class.

Skills we hope to gain:

- Communication
- Teamwork
- Time Management
- Perseverance

Team Members and Skills

- *Jai Burry* – Front-End, Back-End, Quality Assurance
- *Elvis Chiboza* – Client Communication
- *Nicholas Gonzalez* – UI/UX Design, Front-End, Networking
- *Valentine Jingwa* – Front-End, Back-End, Graphic Design
- *Jean-Pierre Nde-Forgwang* – Scheduling/Managing, Front-End, Back-End, Presentation
- *Jonah Scott* – Networking, Quality Assurance, Front-End

Purpose: This contract is designed to clearly define what is expected of each team member, outline our shared responsibilities, and set the rules that will guide how we act and work together during the Software Analysis course. Every member of our team has agreed to follow these rules to help us work well together and achieve our goals.

General Team Rules:

Communication

- Team members are expected to engage in open and transparent communication at all times. This includes sharing relevant project updates, expressing concerns or challenges faced, and providing constructive feedback.
- All interactions should be conducted with respect and professionalism. Team members should listen actively to one another and consider all viewpoints with an open mind before reaching a consensus.
- Team members must acknowledge receipt of emails or messages within 24 hours, even if a full response will take longer. This ensures that all members are aware that their communications have been received and will be addressed.

Meetings

- Team meetings will be held on a weekly basis to ensure consistent communication and alignment on project progress. (Default: during class time)
- The schedule for these meetings will be set in advance and agreed upon by all team members to accommodate everyone's availability.
- Attendance at all team meetings is mandatory to maintain a cohesive and informed team environment.
- In the event that a team member cannot attend a meeting, they are required to notify the Team Leader at least 24 hours in advance, except in cases of emergency.

Work Distribution

- Tasks and responsibilities will be allocated based on an equitable distribution of workload, taking into account each team member's skills, experience, and current commitments to ensure a fair balance.
- The Team Leader, in consultation with the team, will assign tasks. Assignments will be made transparently, and the rationale for task distribution will be communicated to ensure understanding and agreement.

Decision Making

- The team will aim for consensus in decision-making, where all members agree on the decision. This approach fosters collaboration and ensures that all voices are heard.
- If consensus cannot be reached after thorough discussion, decisions will be made by a majority vote. Each team member has one vote, and the option with the most votes will be selected.

Responsibilities:

Team Leader – Jean-Pierre Nde-Forgwang

- Project Planning: Develop detailed project plans that outline key milestones, deliverables, and timelines. Adjust plans as necessary in response to any project shifts.
- Team Coordination: Coordinate the efforts of all team members to ensure that tasks are aligned with project goals and are completed on schedule.
- Conflict Resolution: Identify and address any team conflicts or issues that arise, facilitating a positive and productive work environment.

Quality Assurance - Valentine Jingwa

- Develop Test Plans: Create detailed, structured test plans and cases that cover all aspects of the application's functionality, including edge cases and potential failure points.
- Execute Test Cases: Carry out testing according to the test plans, including functional, system, integration, and user acceptance testing.
- Performance Testing: Assess the application's performance under various conditions to ensure reliability and speed.
- Usability Testing: Evaluate the application for user-friendliness and ensure it meets the usability standards required for the target audience.

Developer – Valentine Jingwa, Jean-Pierre Nde-Forgwang, Elvis Chizoba

- Write, Test, and Debug Code: Develop clean, efficient code based on project requirements. Conduct thorough testing and debugging to ensure functionality and reliability.
- Collaborate on Design: Work closely with the User Experience (UX) and User Interface (UI) designers to ensure that the application's design is effectively translated into functional code.

- Implement Features: Build and implement features and functionalities as outlined in the project specifications

Documentation Specialist – Jean-Pierre Nde-Forgwang

- Ensure Accuracy and Clarity: Verify that all project documentation is precise, clear, and easy to understand. This includes technical documentation, user manuals, and project reports.
- Maintain Documentation Consistency: Ensure consistency in style, format, and terminology across all documentation materials.
- Collaborate with Team Members: Work closely with developers, designers, and other team members to gather necessary information and insights for documentation.

User Interface (UI) Designer – Jonah Scott, Nicolas Gonzalez

- Design the visual elements of the application, including layout, color schemes, and graphics.
- Collaborate with the UX Designer to ensure that the visual design enhances usability.
- Create and maintain a consistent visual identity for the application.

Database Administrator – Jai Burry

- Design the ER diagram for the database in Hackolade.
- Create the database using Compass and MongoDB.
- Collaborate with other team members to ensure the database has all appropriate collections and values.

Timeline:

• Project planning: [September 21st, 2023 - October 20th 2023]
• Development phase: [January 14th, 2024 – March 14th 2024]
• QA and Testing: [March 15th, 2024 – April 15th
2024]
• Submission: [April 19th
2024]
• Tools and Technologies: [SIM Modelling, Mermaid Diagramming, Visual Studio Code, React, UI Kitten, Android Studio, React Native, MongoDB]
• Code Repository: GitHub
• Communication: Discord

Missing Deadlines

- *First Offense:* Written warning
- *Second Offense:* Mandatory extra workload in the next phase
- *Third Offense:* Escalation to the course instructor

Poor Quality Work

- *First Offense:* Opportunity for redo within a 48-hour window
- *Second Offense:* Peer review and extra supervision for the next task
- *Third Offense:* Escalation to the course instructor

Lack of Communication

- *First Offense:* Verbal warning
- *Second Offense:* Written warning
- *Third Offense:* Escalation to the course instructor

Introduction

Vita is a mobile application platform that our team has designed and planned with the objective of revolutionizing how individuals monitor and manage their health data. Our application has designed and planned many features such as health data logging, which can range from diet management, medication intake/output, hydration level monitoring, and a medication scanner utilizing the device's camera to track prescriptions and medications that the user wants to keep a record of, and many other health metrics. With private information the user is entrusting us with, we must ensure our database provides the maximum level of security for our user's peace of mind, and to make sure that all of the data processed by our application is received, stored, and secured at all times. This testing strategy report solidifies our approach to ensuring the applications' accessibility, security, and performance to uphold our expectation of the highest standard of quality.

Project Scope and Requirements Review

Vita has a broad range of features and functionalities that we believe are critical and interlinked with each other to deliver a seamless and reliable experience for the user. This includes:

User Authentication

- Secure login mechanisms to protect user information and two-factor authentication via a fingerprint scanner, face identification, or user-set security questions if enabled.

Health Data Logging

- Input and Output of health metrics like steps, hydration, medication, and diet. Process the information into the correct categories and format for display on our health history chart

Health Data Viewing

- Responsive and accurate charting capabilities that format user inputs to be displayed in user-defined time intervals. Updating charts and formatting information as soon as a user inputs a metric into our system.

Device Connectivity

- Integrating and supporting external health monitoring devices like the Fitbit, Apple Watch, and Garmin smartwatches. Integrate this technology and information into our application while also adding data input from those devices to be formatted.

Account Management

- Creation of a personal user account with saved accessibility settings, as well as creation of subaccounts like a child account that can be monitored and managed by the parent user with easy access.

Data Security/Sharing

- A strong and secure database that processes, stores, as well as backs up user information. As well as including a secure option of sharing or printing out data if the user requires it.

Our team believes that our success on this application depends on a seamless and flawless execution of all of these categories working together. By doing this, we ensure our priorities and develop a good testing format to ensure our products' success.

Magnitude and Sources of Development Risk

Our team has decided that our risks and the magnitudes associated with them will be prioritized strictly on the magnitude of the potential issue and the effect it will have on our application if overshadowed.

Technical Complexity - High Magnitude

With third-party device integration as well as our scanner, having errors and issues in these categories could result in a loss of data, app crashes, and compatibility issues. We believe this will result in extreme user dissatisfaction, and delay the success of our app by not having key features promised in our planning which is why we have labeled it as a high-magnitude issue.

Data Security - Very High Magnitude

Handling and storing an individual's health data will require us to implement advanced security measures and safeguards to ensure such sensitive data is kept only where the user wants. We have labeled this as our number one highest magnitude risk factor in building our application, error to ensure quality security can lead to a loss of user data or even stolen private information by a security breach which can lead us to legal and financial problems, and tarnish our reputation in the space.

User Interface/Accessibility - Medium Magnitude

With expecting users from many different cultures and ages we need to ensure customer satisfaction and make our app accessible and customizable for all users. We have marked this as a medium-magnitude issue as we need to make sure all customers willing to try out our product at launch will have a polished application and be satisfied.

Completion Criterion

Our testing will be considered completed once we have achieved this:

- 95% of the code will be functional and working through all pre-determined tests
- Critical Errors/Defects in code or logic have been reduced under 1% in total
- All high and very high-magnitude issues have been successfully resolved

Management of Testing in Vita

Testing will be managed equally throughout the team through all phases and cycles of development with members of the team focussing on testing throughout each goal completed, as well as two weekly team meetings designed to ask questions and provide insights on current testing results and new findings. We will be planning to have integration, system, and acceptance testing at their respective phases of development.

Optimal Testing Timing

Vita has set testing frameworks and timeframes to ensure our product is always being developed at maximum efficiency to ensure a quality product is returned to our stakeholders at the desired time. Our testing phases include:

Early Phase

- Focusing on unit and integration testing to ensure all errors and defects are spotted and resolved early to avoid pile-ups in the later phases.

Middle Phase

- Once completing our initial draft of the application and are satisfied with our app as a whole, we will start system and usability testing to fully complete and optimize our UI and system integration.

Final Phase

- In our final phase of development, we expect to have a fully running and polished application in which we will start performance and security testing to ensure our product can keep data safe and can ensure more scalability.

Cost of Failure

Vita's cost of failure could be from multiple different scenarios. Delays on the project, and insufficient security resulting in loss or stolen data, which includes the complete loss of reputation and trust in our users, we expect losses to be substantial, estimating over $500,000, and possibly over $1M. This further reinforces the need for our team to proactively test our application ensuring we have information on all possible defects.

Risk Reduction Through Testing

Positive Testing

- Vita will utilize positive testing by essentially being our own end users. We will be using our application in the eyes of a user and inputting and navigating our app in normal circumstances. Logging in, adding data to an account's health log, and ensuring values are returned to the system and stored correctly. This testing will ensure our application works under normal operation conditions.

Negative Testing

- Negative testing will be utilized by our team going into the application and trying to essentially break it. This is where we will attempt to input wrong values, and attempt unexpected user actions that would be abnormal to our system, in order to test how our application handles those errors, crashes, and incorrect formats that may arise during the testing. This will allow us to gain more insight into potential areas of weakness and can help us enhance our application to handle these types of inputs.

Prioritizing Risks

Vita will be using a risk hierarchy that will help us understand and develop a list of the highest severity and likelihood of occurring during the operation. The risks that have the highest magnitude of severity as well as the ones we find have the most common chance of happening will be focussed on and resolved first. This will allow us to have a detailed and ordered list that we can spend the most time and allocate more resources towards fixing. Once the issue is resolved we remove it from the list and move on to the next until the list is cleared. This prioritization of risk handling allows us to be as efficient as possible and remove significant risks as soon as they are identified.

Statistical Analysis of Defects' Arrival Patterns

Using statistical analysis when an error or defect arises and is recorded, analyzed, and resolved by the team, we will utilize those test results and mark down why they occurred, what resulted from it, and how we resolved it as well as what phase of development and area the issue came into play. Gathering information on this will help us notice any patterns that may form, or a certain way our code is implemented throughout the application will help us adapt and avoid issues in the future, and gain more insight on how we are able to mitigate issues with the highest efficiency. Using statistical analysis we will also be able to form stronger timeframes for testing having previous data and possible relative issues and how long they took to complete. Once we are recording fewer and fewer issues throughout our system we will know our application is reaching strong stability and readiness for release.

Customers Using the System

Vita will be using customer and stakeholder feedback on the application to improve and broaden our range for testing based on how our test users use the application. After collecting the data of how a typical user will interact with our system we mark patterns and trends in each user and how they navigate our app on a daily basis. Using these patterns we can refine and focus on creating test cases on examples in real-world scenarios. Learning how real users will interact with our app before deployment is essential for us and extremely useful in adding new test cases and focusing on regular user interactions.

Testing Strategies

Vita believes all testing strategies outlined in the course are necessary to be used in each of our development cycles to ensure quality and efficiency. The reason we chose these is:

Static Testing

- This will be used in our early development cycles to ensure our code is correct logically, with correct syntax, and catch potential security issues before we execute the code. This will also help us keep on track with our design layout and plan to avoid going off track.

White Box Testing

- This is where we will test our system to ensure the structure of our application responds well to each option and execution that our app will support. This can help us make sure each path that the app can take is solid and reliable.

Black Box Testing

- This type of testing will be used by us during our late stages of development to test our UI, and how the app functions as if we were end users without too much focus on the code structure. This is where we will reflect on our design and expectations from stakeholders to make sure we have met all requirements.

Performance Testing

- This type of testing will be used to stress test our application and see how it responds under lots of loads, in our expectation of having many users and traffic constantly flowing through our app. We will be processing and storing lots of mock data to see if any vulnerabilities or app crashes happen. This will help us understand and improve our performance and scalability and keep our application as responsive as possible.

Peer Assessment

	<JEAN PIERE>	<NICO GON>	<JONAH>	<VALENTINE>
Motivation and engagement in assignment	3/3	3/3	3/3	3/3
Timely submission of deliverables	3/3	3/3	3/3	3/3
Effective communication and problem-solving	3/3	3/3	3/3	3/3
Other comments for your peers				