Optimal State

Software Project Management Plan (SPMP)

OPTIMAL

State of Living™

Project Manager/Computer Engineer: Mohammed Gharaibeh

Assistant Project Manager/Computer Engineer: Majd Alissa

Computer Engineers: Alaa Abushawer & Yousef Hafnawi

Table of Contents

1. Overview

1.1 Project summary

1.1.1 Purpose, scope, and objectives

1.1.2 Assumptions and constraints

1.1.3 Project deliverables

1.1.4 Schedule and budget summary

1.2 Evolution of the plan

2. References

3. Definitions

4. Project organization

4.1 External interfaces

4.2 Internal structure

4.3 Roles and responsibilities

5. Managerial process plans

5.1 Start-up plan

5.1.1 Estimation plan

5.1.2 Staffing plan

5.1.3 Resource acquisition plan

5.1.4 Project staff training plan

5.2 Work plan

5.2.1 Work activities

5.2.2 Schedule allocation

5.2.3 Resource allocation

5.2.4 Budget allocation

5.3 Control plan

5.3.1 Requirements control plan

- 5.3.2 Schedule control plan
- 5.3.3 Budget control plan
- 5.3.4 Quality control plan
- 5.3.5 Reporting plan
- 5.3.6 Metrics collection plan

5.4 Risk management plan

5.5 Closeout plan

6. Technical process plans

6.1 Process model

6.2 Methods, tools, and techniques

6.3 Infrastructure plan

6.4 Product acceptance plan

7. Supporting process plans

7.1 Configuration management plan

7.2 Verification and validation plan

7.3 Documentation plan

7.4 Quality assurance plan

7.5 Reviews and audits

7.6 Problem resolution plan

7.7 Subcontractor management plan

7.8 Process improvement plan

Software Project Management Plan (SPMP)

1. Overview

1.1 Project Summary

The Optimal State project aims to develop an Android application that helps users monitor and improve their mental health through assessments, guided exercises, and history tracking. The app also provides tools for providers to manage clients and view their mental health records, with data stored and managed via Firebase.

1.1.1 Purpose, Scope, and Objectives

- **Purpose**: To develop the Optimal State Android application that supports users in monitoring and improving their mental health through assessments and exercises while enabling providers to manage and track client progress.
- **Scope**: The project includes creating an Android app with key features such as user authentication, mental health assessments, guided exercises, history tracking, and a provider interface for client management. The application will utilize Firebase for real-time database management.
- **Objectives**:
  1. Create a user-friendly interface for individual users and providers.
  2. Implement functional modules for assessments, exercises, and history tracking.
  3. Enable secure and efficient data management through Firebase
  4. Deliver prototypes iteratively, refining features based on feedback.

1.1.2 Assumptions and Constraints

- **Assumptions:**
  1. All team members will actively participate and meet deadlines.
  2. The Firebase database and services will remain operational and reliable.
  3. Users will have Android devices with internet connectivity to use the application.

  4. The development environment (Android Studio) and tools will remain stable and compatible.

- **Constraints:**
  1. Limited storage and bandwidth on the free Firebase tier (1GB storage, 10GB downloads).
  2. The application is designed exclusively for the Android platform and cannot be ported to other systems in its current phase.
  3. The team must complete the project within the semester timeline.

1.1.3 Project Deliverables

- **Android Application:**
  1. A fully functional Android app with the following features:
     a) User authentication (login and account creation).
     b) Self-assessment and mental health tracking.
     c) Guided exercises for mental health improvement.
     d) History tracking for users.
     e) Provider interface for client management and history review.
- **Documentation:**
  1. Software Requirements Specification (SRS).
  2. Software Project Management Plan (SPMP).
  3. Maintenance and user manuals.

- **Source Code:**
  1. Well-documented source code uploaded to a version control system (e.g., GitHub).
- **Prototypes:**
  1. Prototype 1: Basic app functionality, including login and assessment interfaces.
  2. Prototype 2: Full functionality with remove client & notification features .
- **Deployment:**
  1. A publishable version of the application, ready for release on the Google Play Store.

1.1.4 Schedule and Budget Summary

- **Schedule & Budget Summary:**
  1. The project has no allocated budget. Development will utilize free tools and services, including:
     a) Android Studio for application development.
     b) Firebase (free tier) for database and authentication services. Any additional costs, such as publishing on the Google Play Store, will be minimized or simulated for educational purposes.

1.2 Evolution of the Plan

- **The project plan will evolve iteratively as follows:**
  1. **Initial Planning:**
     a) The team met with stakeholders to define project requirements and expectations based on the SRS document. Key features and deliverables were outlined, and the timeline was established.
  2. **Prototype Development:**
     a) Prototype 1: Focuses on developing the basic app structure, including user authentication, assessment functionality, and guided exercises.

b) Prototype 2: Builds on Prototype 1 by integrating notifications Feature & Remove Client.

3. **Testing and Feedback:**
   a) Each prototype will undergo unit, integration, and system testing. Feedback from stakeholders will guide refinements and adjustments to ensure the application meets requirements.
4. **Final Delivery:**
   a) The completed application, along with documentation and source code, will be submitted, and a final presentation will demonstrate its functionality and usability.
5. **Maintenance and Future Plans:**
   1. Documentation and code will be prepared for ease of maintenance and potential future enhancements.

2. References

- Android Studio: https://developer.android.com/studio
- Firebase : https://firebase.google.com
- IEEE Software Requirements Specification (SRS) Template: https://standards.ieee.org
- Project SRS Document: Detailed requirements and functionality for the Optimal State application.

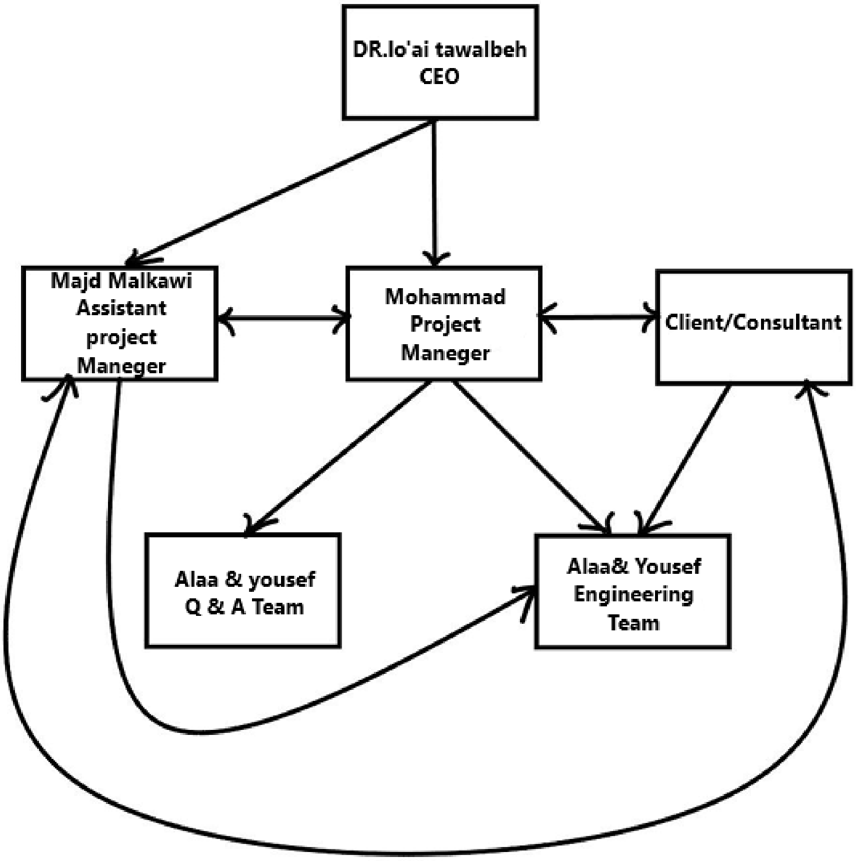
3. Definitions

- **Android**: A mobile operating system used to develop the Optimal State application.
- **Firebase**: A cloud-based platform offering real-time database and authentication services used to store and manage user data.
- **SRS (Software Requirements Specification):** A document outlining the functional and non-functional requirements of the project.
- **Prototype:** An early version of the application demonstrating basic functionality and user interfaces.
- **User:** An individual utilizing the application for mental health monitoring and improvement.
- **Provider:** A healthcare professional using the application to manage and review clients' mental health data.
- **Assessment:** A feature in the application that allows users to evaluate their mental health state.
- **Guided Exercises:** Exercises and techniques offered within the application to help users improve their mental well-being.

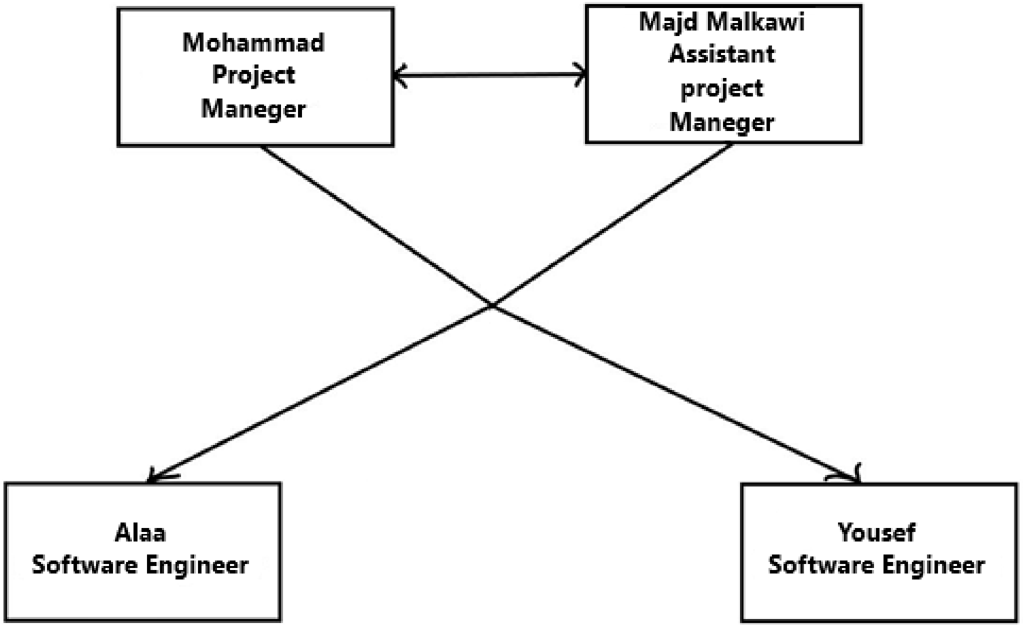
- **History Tracking:** A feature enabling users and providers to review past mental health assessments and progress.
- **UI/UX:** User Interface and User Experience, focused on designing intuitive and visually appealing application interfaces

4. Project Organization

4.1 External Interfaces



4.2 Internal Structure



4.3 Roles and Responsibilities

- Mohammad Gharaibeh(Project Manager): Liaison with stakeholders, ensures alignment with project objectives.

- Majd Malkawi (Assistant Project Manager): plays a critical supporting role in the successful execution of the Optimal State project

- Alaa abushawer (Software Engineer): Handles app development and Firebase integration.

- Yousef Hafanawi (UI/UX Designer): Designs and tests user interfaces for accessibility and aesthetics.

- Alaa abushawer (Quality Assurance Specialist): Conducts testing and prepares bug reports.

5. Managerial Process Plans

5.1 Start-Up Plan

Initial activities include:
- Team formation and role assignment.
- Requirement analysis based on the SRS.
- Environment setup (Firebase, Android Studio).

5.1.1 Estimation Plan

- Collaborate with the client to define and confirm all application specifications and requirements.
- Analyze and explore the existing codebase to facilitate the integration of new features into the application.
- Monitor the team's overall progress to ensure alignment with project goals and timelines.
- Establish clear guidelines and deadlines for the development and delivery of prototypes 1 and 2.

5.1.2 Staffing Plan

Project staffing was determined based on each team member's skills, expertise, and availability. The roles were allocated as follows to ensure efficient project execution:

- **Project Manager:** Mohammed Gharaibeh, responsible for overseeing project progress, facilitating communication with stakeholders, and ensuring adherence to timelines.
- **Assistant Project Manager:** Majd Alissa, assisting in coordination, documentation, and stepping in as needed to support project management tasks.
- **Software Engineers:**
  - **Alaa Abushawer:** Focused on developing core application features, including user authentication and data management.
  - **Yousef Hafnawi:** Specialized in integrating Firebase services and implementing user history tracking features.
- **Quality Assurance Specialist (Alaa Abushawer):** Conducting rigorous testing to identify and resolve bugs, ensuring the application meets quality standards.

5.1.3 Resource Acquisition Plan

The resources required for the development of the project will be obtained as follows:

1. **Software Tools and Platforms:**
   - **Andriod Studio:** The primary Integrated Development Environment (IDE) for Android application development. It is open-source and free to download from the official website.
   - **Firebase database:** Utilized for real-time database management and user authentication. The project will use Firebase's free tier, which offers sufficient features for the development phase.
2. **Hardware Resources:**
   - Personal computers or laptops of team members with sufficient specifications to run Android Studio and related tools.

3. **Collaboration Tools:**
    - **GitHub:** A version control platform for managing the project's source code and facilitating team collaboration.

4. **Knowledge Resources:**
    - Online tutorials and guides for Firebase and Android development

5.1.4 Project Staff Training Plan

The team will undergo training sessions to familiarize themselves with Firebase, Android Studio, and Agile methodologies. Tutorials and resources will be shared to ensure that all members are equipped to contribute effectively.

Training will be conducted during the first two weeks of the project.

5.2 Work Plan

5.2.1 Work Activities

The project's work plan is divided into specific units to ensure structured progress and timely delivery.

- **Prototype 1 – Core Navigation Drawer Development:**
  The first prototype will focus on establishing the foundational Navigation Drawer as the main interface for user navigation. Key functionalities will include login and registration, providing users with access to their personalized app experience. The Navigation Drawer will feature links to essential sections such as "Take Assessment", "View Exercises" and "Settings", "View History" ,and more feature for provider such as "add client","settings","view client history",designed to ensure a user-friendly and intuitive experience. The interface will incorporate a vertical layout with clear options and icons. All team members will collaborate to design, develop, and thoroughly test this feature.

  Prototype 2 – Advanced Integration and Expansion:
  The second prototype will build upon the functionalities introduced in Prototype 1, integrating advanced features for both users and providers. Key enhancements include" remove client", "notifications feature" An interactive tutorial will be introduced to guide first-time users in utilizing the Navigation Drawer.

- **Documentation:**
  To ensure the application is maintainable and easy to understand, all team members will focus on thorough documentation. This includes detailed comments within the codebase, the creation of comprehensive user manuals, and regular updates to technical documentation to capture the progress and changes made during the project.

5.2.2 Schedule allocation

12/1/2024	12/2/2024	Prototype #1 Development (Login and Registration)
12/5/2024	12/8/2024	Prototype #1 Development (Home Screen Navigation)
12/10/2024	12/12/2024	Prototype #1 Development (Take Assessment button)
12/14/2024	12/16/2024	Prototype #1 Development (View Exercises button)
12/18/2024	12/20/2024	Prototype #1 Development (View History button)
12/22/2024	12/24/2024	Prototype #1 Development(Settings button)
12/26/2024	12/26/2024	Prototype#1 development (Provider Home screen)
12/27/2024	12/27/2024	Prototype#1 development (Add client, View Client History)
12/28/2024	12/28/2024	Prototype#1 development (settings button)
12/29/2024	12/29/2024	Prototype#2 development (notifications feature)
1/1/2025	1/1/2025 Happy new year	Prototype#2 development (Remove Client)

5.2.3 Resource allocation

All team members will have access to the necessary resources for the successful development and completion of the project. These resources include:

- **Development Tools**:
    - **Android Studio**: The primary Integrated Development Environment (IDE) for Android application development.
    - **Firebase Services**: For authentication, database management, and real-time data synchronization.
- **Hardware**:
    - Personal computers or laptops capable of running Android Studio and Firebase integrations.
    - Access to high-speed internet for seamless collaboration and cloud services.
- **Collaboration Tools**:
    - **GitHub**: For version control, collaboration, and tracking code changes.
    - **Communication Platforms**: Slack or email for team communication and coordination.
- **Knowledge Resources**:
    - Tutorials and online documentation for Android development and Firebase services.
    - Support from project supervisors or mentors for technical guidance.
- **Testing Devices**:
    - Android devices for testing and debugging the application features during development.

These resources are distributed equally among the team members to ensure fair and efficient collaboration. The team is committed to optimizing resource usage to stay aligned with project objectives and timelines.

5.2.4 Budget allocation

This project has been designed to utilize freely available resources and services, ensuring no monetary budget is required.

5.3 Control Plan

5.3.1 Requirements Control Plan

- **Team Meetings**:

    - **Scheduled Meetings**: All team members must attend weekly meetings to review project progress, discuss challenges, and refine requirements.
    - **Ad-Hoc Meetings**: Additional meetings will be arranged outside scheduled times as needed to address critical issues or client feedback.

- **Documentation and Adherence**:

    - All team members are responsible for documenting their work thoroughly in alignment with the guidelines set forth in the Software Requirements Specification (SRS).

  - Documentation must include detailed comments in code, updated project notes, and changes recorded in the version control system (GitHub).

- **Change Management**:

  - Any changes or updates to requirements will be communicated by the Project Manager to all team members.
  - Changes must be reviewed during team meetings to assess their impact on the project schedule, resources, and deliverables.

- **Review and Resolution**:

  - **Challenges and Issues**: Unforeseen technical issues, missed deadlines, or client requests will be escalated to the managing team (Project Manager and Assistant Project Manager).
  - The managing team will assess the issue, consult with relevant stakeholders, and propose solutions to ensure minimal disruption to the timeline.

5.3.2 Schedule Control Plan

To ensure the timely delivery of project milestones, the following schedule control measures will be implemented:

- Task Monitoring and Tracking: The Project Manager will monitor the progress of each task and ensure adherence to deadlines outlined in the work plan. Tasks will be tracked using a shared project management tool.

- Regular Progress Reviews: Weekly meetings will be held to review progress, identify potential delays, and discuss any adjustments needed to meet the project timeline.

- Contingency Planning: A buffer period will be built into the schedule to accommodate unforeseen delays or challenges. The team will prioritize critical tasks to ensure project objectives are met within the semester timeline.

- Communication and Accountability: Team members will be required to report on their task progress during weekly meetings. Any delays or issues will be escalated to the Project Manager for resolution.

- Additional Work Sessions: If necessary, additional team meetings or work sessions outside standard class times will be arranged to meet project milestones.

By adhering to these measures, the team will ensure that all tasks are completed within the established timeframe, enabling the successful delivery of the project.

5.3.3 Budget Control Plan

No budget has been allocated for the Optimal State project. The team will utilize free resources, including Android Studio and Firebase (free tier), to maintain cost efficiency. Any unexpected expenses will be evaluated and minimized to avoid financial strain on the project.

5.3.4 Quality Control Plan

To ensure the project meets its quality standards and aligns with the requirements outlined in the Software Requirements Specification (SRS), the team will implement the following quality control measures:

1. **Regular Quality Checks**:
   - Weekly reviews will be conducted by the project manager and the development team to evaluate the progress and functionality of the application.
   - Each module or feature will be tested upon completion to verify it meets specified requirements.
2. **Testing Strategy**:
   - Unit testing for individual components to ensure correct functionality.
   - Integration testing to verify seamless interaction between different components.
   - System testing to evaluate the overall performance and reliability of the application.
3. **Bug Tracking and Resolution**:
   - Any identified issues or bugs will be documented and assigned for resolution promptly.
4. **Final Quality Review**:
   - Before deployment, a final review will be conducted to ensure all requirements are met, and the application is stable and user-friendly.

Through these steps, the team will maintain high-quality standards and deliver a product that satisfies client and user needs.

5.3.5 Reporting Plan

The project manager will maintain consistent communication with the supervisor, Dr. Lo’ai Tawalbeh, through bi-weekly reports. These reports will include summaries of team progress, and updates on application development to ensure transparency and alignment with project objectives.

5.3.6 Metrics Collection Plan

The managing team will track weekly task completion rates for all developers to ensure the project remains on schedule. Metrics such as lines of code written, bug resolutions, and task completion will be monitored. Regular evaluations will ensure the code meets efficiency and quality standards as defined in the SRS.

Productivity & Quality Measure

Productivity Metric: 20 lines of code per hour
Quality Metric: 40 faults per 350 lines of code

Alaa Abushawer: Conducted detailed video conferencing with the team regarding the implementation of Firebase integration and user authentication. Provided step-by-step guidance on coding the login and registration features, implementing approximately 20 lines of code.

Majd Alissa: Spent 1 hour per week, in addition to meeting and lab hours, working on the development of the mental health tracking interface and history module, including Firebase database queries. Contributed 100 lines of code.

Yousef Hafnawi: Dedicated significant time to researching and debugging issues in the user interface (UI) design and ensuring compatibility with Android Studio. Implemented 100 lines of code while refining the app's navigation and structure.

Alaa Abushawer (Quality Assurance Role): Spent considerable time developing and testing guided exercise features. Focused on identifying and resolving bugs in the assessment module, contributing 100 lines of code.

Mohammed Gharaibeh: Spent extensive time on project management, including designing and implementing the assessment logic and reviewing the Firebase database setup. implement 100 lines of code for the provider interface.

5.4 Risk Management Plan

Effective risk management is essential to ensure the success of the Optimal State project. The following measures will be implemented to manage and mitigate risks:

o *Regular Team Meetings*: The team will meet weekly to identify potential risks, discuss their impact, and determine mitigation strategies.

o *Deadline Monitoring*: Each task's progress will be tracked closely to ensure that delays are identified early and addressed proactively.

o *Resource Allocation*: Resource requirements will be assessed regularly to avoid shortages and ensure the availability of tools and infrastructure necessary for project development.

o *Communication Protocols*: Clear communication channels will be maintained among team members, stakeholders, and the project supervisor to address issues as they arise.

o *Risk Mitigation Plans*: For identified risks, a mitigation plan will be developed, including contingency actions to minimize their impact on the project timeline and deliverables.

By adopting these measures, the team aims to minimize risks and ensure the project stays on track.

5.5. Closeout plan

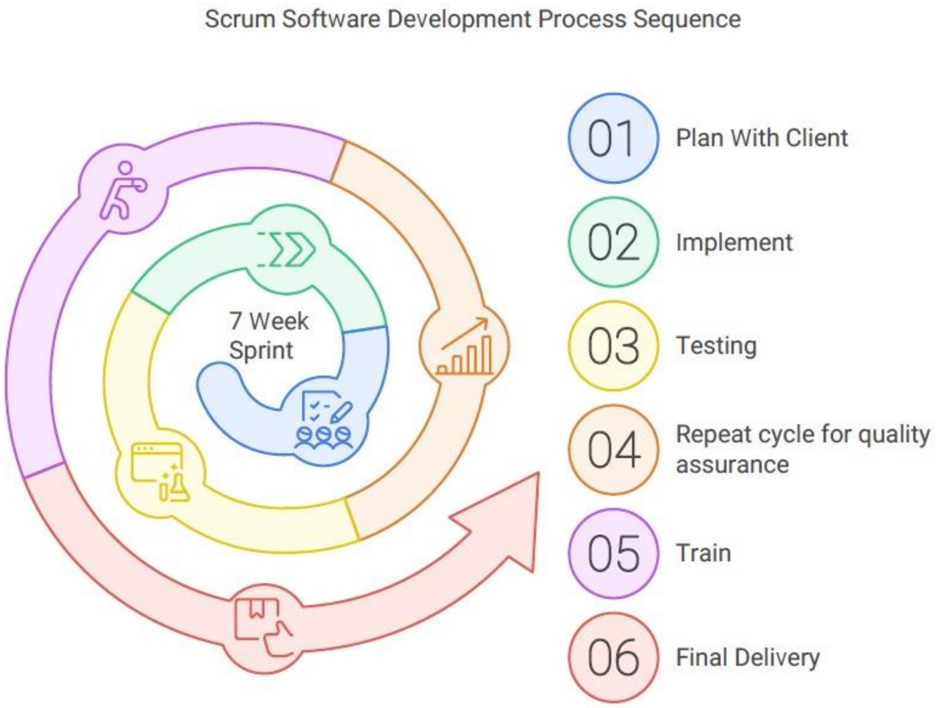
To ensure a smooth and organized project conclusion, the following closeout activities will be undertaken:

1. **Deliverable Submission**:
   - All project deliverables, including the source code, documentation (e.g., Software Requirements Specification (SRS), Software Project Management Plan (SPMP), Maintenance Manual, and User Guide), will be uploaded to a centralized repository on GitHub or an equivalent platform (e.g.,elearning).
2. **Final Presentation**:
   - The team will prepare and deliver a comprehensive presentation showcasing the application’s features, functionality, and technical achievements.

3. **Handover to Stakeholders**:
    - The final version of the application will be made accessible to the stakeholders for their review.
4. **Team Evaluation and Documentation**:
    - A retrospective meeting will be conducted to discuss the project's successes, challenges, and lessons learned.
    - These insights will be documented to improve processes for future projects.

By following this plan, the team will ensure a professional and complete project closeout, leaving all deliverables and documentation in a state that supports future maintenance and use.

6.1 Process Model


Scrum Software Development Process Sequence
7 Week Sprint
01 Plan With Client
02 Implement
03 Testing
04 Repeat cycle for quality assurance
05 Train
06 Final Delivery

The Scrum model is essentially divided into six components: train, plan, implement, test, repeat x times, and deliver. For our purposes, the OPTIMAL State Living team will first train. Each member is responsible for familiarizing themselves with the necessary tools and frameworks to develop the app.

Next is planning with the stakeholder. Before development can begin, the team must consult with the stakeholder to clarify the expectations, features, and procedures for the app. This ensures alignment with the project's goals and requirements as outlined in the SRS.

After planning, the team must implement the designs according to the stakeholder's specifications. During implementation, each team member is responsible for coding their assigned components of the application.

After implementation, testing is carried out. Each team member will test the components they developed to ensure they function correctly and integrate seamlessly with the rest of the system. The testing process includes unit, integration, and system testing.

If further adjustments or improvements are needed, the Scrum cycle will repeat. This iterative process allows the team to address concerns and enhance the software for better quality and performance. The cycle may be repeated as many times as needed to achieve the desired quality, starting from planning with the stakeholder.

The final step is delivering the complete product, assuming no additional iterations are required. All these procedures will be conducted within a seven-week timeframe to meet the project deadlines and objectives.

6.2 Methods, Tools, and Techniques

Methods:

- The **Incremental Development Model** will be used to ensure the project is developed in iterative cycles. Features will be implemented and tested in increments, allowing for gradual refinement and early feedback from stakeholders. This method ensures flexibility in accommodating changes and improvements throughout the development process.

Tools:

1. **Development Tools**:
    - **Android Studio**: The primary Integrated Development Environment (IDE) for building, testing, and debugging the Android application.

    - **Firebase**: A platform for real-time database management, authentication, and backend services.
2. **Version Control**:
    - **Bitbucket (or GitHub)**: A version control system used to manage the source code, track changes, and enable collaborative development.
3. **Collaboration Tools**:
    - **Communication Platforms**: Slack, email, or equivalent tools for seamless team communication.

Techniques:

1. **Client Interaction**:
    - Prototypes and feature demonstrations for iterative validation of the product's alignment with client expectations.
2. **Team Collaboration**:
    - Weekly team meetings to assess progress, resolve challenges, and plan next steps.
    - Pair programming and code reviews to ensure high-quality code development and knowledge sharing.

3. **Agile Practices**:
    - Breaking down the project into sprints, each focused on delivering specific features or functionality.
    - Continuous integration and testing to identify and resolve issues early in the development process.

6.3 Infrastructure Plan

The application infrastructure relies on Firebase for database and authentication services. All team members will access a shared GitHub repository for collaboration. Android Studio is the primary development environment, with no additional infrastructure requirements.

6.4 Product Acceptance Plan

The product acceptance plan ensures that the final application meets the stakeholder's expectations and adheres to the requirements outlined in the Software Requirements Specification (SRS). The following steps will be undertaken:

1. **Stakeholder Involvement**:
    - The stakeholder will be actively involved throughout the development process, participating in regular progress reviews and providing feedback on prototypes.
2. **Quality Assurance (QA) Testing**:
    - The QA team will perform rigorous testing on all features, including unit, integration, and system testing, to identify and address any issues before final delivery.

3. **Prototypes and Iterative Feedback**:
    - The project will be delivered in two main prototypes:
        - **Prototype 1**: Basic functionalities, including login, assessments, and guided exercises.
        - **Prototype 2**: Advanced features, such as remove client and notifications feature.
    - Feedback from the stakeholder on each prototype will be used to refine and enhance the application.
4. **Final Acceptance**:
    - Upon completion of the final iteration, the application will be presented to the stakeholder for final review and approval.
    - The stakeholder will verify that all features have been implemented as specified and meet their expectations.
5. **Deliverables Handover**:
    - The final product, along with all associated documentation (e.g., SRS, SPMP, Maintenance Manual, User Guide), will be handed over to the stakeholder.
    - Any additional feedback provided by the stakeholder during the final review will be documented for future development phases.

7.1 Configuration Management Plan

GitHub will be used for configuration management, allowing the team to track changes, resolve conflicts, and maintain version control. The repository will be regularly updated and reviewed to ensure consistency.

7.2 Verification and Validation Plan

The verification and validation (V&V) process ensures that the application meets the requirements specified in the Software Requirements Specification (SRS) and functions as intended. The following steps will be taken:

1. Verification Process

Verification focuses on ensuring that the software is built correctly according to the specified requirements. The following types of testing will be conducted:

1. **Unit Testing**:
    - Each team member is responsible for testing their assigned software components.
    - Unit tests will validate the functionality of individual components or modules to ensure they work as expected in isolation.
    - Issues or bugs identified during this stage will be documented and resolved before moving to integration testing.
2. **Integration Testing**:
    - Once individual components are verified, they will be combined and tested as a whole to check for compatibility and interaction issues.

    - All team members will participate in integration testing to identify and resolve interdependencies and communication errors between modules.
3. **System Testing**:
    - The application will be tested as a complete system to verify that it meets the functional and non-functional requirements outlined in the SRS.
    - This stage will ensure the overall performance, reliability, and usability of the application.

2. Validation Process

Validation focuses on ensuring the application fulfills its intended purpose and satisfies stakeholder expectations. The following activities will be performed:

1. **Stakeholder Involvement**:
    - The stakeholders will participate in periodic reviews to validate prototypes and provide feedback.
    - Feedback will be incorporated into subsequent iterations to ensure the application aligns with client expectations.
2. **User Acceptance Testing (UAT)**:
    - End users, including stakeholders and testers, will evaluate the application to confirm that it meets their needs and expectations.
    - Any issues or requested changes identified during UAT will be documented and prioritized for resolution.

3. Bug Tracking and Resolution

- Bugs or errors found during any phase of testing will be logged in a bug tracking tool (e.g., GitHub Issues) for tracking and resolution.
- The status of each bug (e.g., new, in progress, resolved) will be monitored to ensure timely fixes.

4. Iterative Testing and Scrum Cycle

- If critical errors or unmet requirements are identified during system testing or validation, the development process will return to the appropriate stage of the Scrum cycle.
- Testing will repeat from unit testing to system testing to validate corrections and ensure quality.

7.3 Documentation Plan

The team will produce the following documents:
- Software Requirements Specification (SRS): Detailing application functionality and constraints.
- Software Project Management Plan (SPMP): Documenting project processes and management.

7.4 Quality Assurance Plan

The Quality Assurance Specialist will ensure that all deliverables meet the required standards through regular testing and adherence to the SRS. Weekly reviews will ensure ongoing quality control.

7.5 Reviews and Audits

Regular reviews will be conducted by the team and project supervisor to identify and resolve issues early. Audits will be performed at each milestone to ensure compliance with project goals.

7.6 Problem Resolution Plan

Team members will report issues to the Project Manager, who will prioritize and assign tasks for resolution. Critical issues will be escalated to the Assistant Project Manager for immediate attention.

7.7 Subcontractor Management Plan

No subcontractors are involved in this project. All tasks are managed internally by the project team.

7.8 Process Improvement Plan

The team will conduct retrospectives after each prototype delivery to identify areas for improvement. Documentation will be updated to reflect lessons learned, ensuring future projects benefit from this experience.