Question 1

a. Customer problem: Slow and unreliable internet connection is preventing the adoption of an online payments platform in rural areas.
b. How do you know it is a problem? The problem has been reported by several rural communities during surveys and market research. It is also reflected in the low adoption rates of the online payments platform in these areas.
c. Why it is critical to solve the problem: Online payments are becoming increasingly important for commerce, and the lack of access to this service hinders economic growth and development in rural areas. Solving this problem will allow people in rural areas to participate in the digital economy and access new opportunities.
d. How will you measure the success of the solution? Success will be measured by the increase in adoption and usage rates of the online payments platform in rural areas. Other metrics include internet speed and reliability, customer satisfaction, and economic growth in these areas.

Question 2

a. Solution Diagrams

Diagram 1 - System Architecture

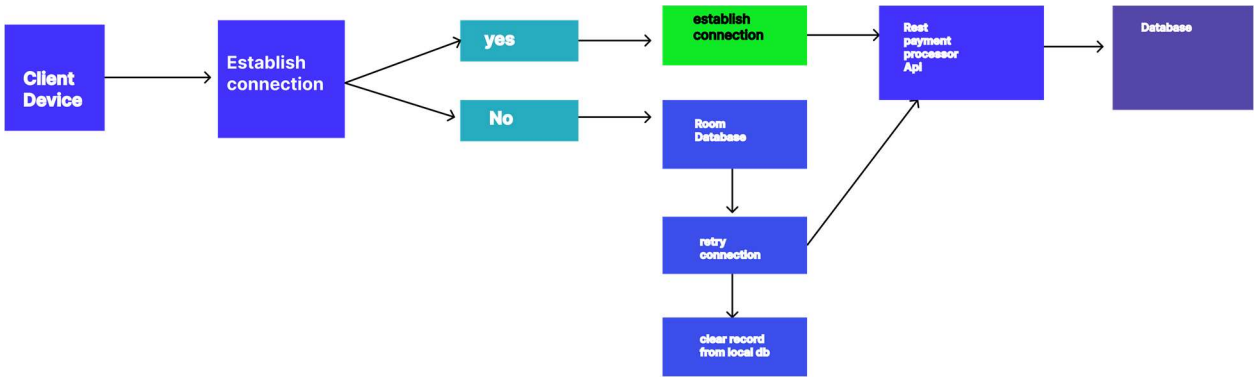

Diagram 2 - Project Management Workflow

1. Research

- Research best practices and techniques for optimizing mobile apps for low-bandwidth and offline usage.
- Investigate potential libraries and frameworks that can be used for optimizing app performance and data usage.
- Document findings and recommendations in a report.
- Identify areas in the mobile app that can be optimized for low-bandwidth and offline usage.
- Prioritize optimization tasks based on impact and feasibility.
- Develop a plan for implementing the optimizations.
- Implement optimizations in the mobile app, including:
  - Minimizing app size
  - Reducing data usage
  - Implementing caching and pre-fetching mechanisms
  - Optimizing app performance
- Conduct preliminary testing of the app in a lab environment to evaluate the effectiveness of the optimizations.
- Identify any technical issues that need to be addressed.
- Update the optimization plan as needed based on the results of the testing.

2. Proof of cooncept

- Week 1:
  - Day 1:
    - Review the research and recommendations from Card 1 to ensure that the proposed solution is feasible and effective.
    - Develop a plan for building the proof of concept.
  - Day 2-3:
    - Set up the development environment and necessary tools.
    - Create a basic prototype of the solution to test its viability.
    - Document the development process and any technical issues that arise.
  - Day 4-5:
    - Conduct initial testing of the prototype to evaluate its effectiveness.
    - Identify any technical issues that need to be addressed.
    - Update the development plan as needed based on the results of the testing.
- Week 2:
  - Day 1-2:
    - Refine the prototype based on the results of the initial testing.
    - Implement any additional features or improvements identified during testing.
  - Day 3-4:
    - Conduct comprehensive testing of the prototype to identify any issues and bugs.
    - Gather feedback from users and identify any areas for improvement.
  - Day 5:
    - Make necessary updates to the prototype based on the feedback and issues identified during testing.
    - Conduct final testing of the prototype to ensure stability and reliability.

3. Integrate Online Payments Platform with Payment Infrastructure

- Week 1:
  - Day 1:
    - Identify potential payment gateways, banks, and other financial institutions to integrate with.
    - Document findings and recommendations in a report.
  - Day 2-3:
    - Develop a plan for integrating the online payments platform with the identified payment infrastructure and systems.
    - Prioritize integration tasks based on impact and feasibility.
    - Define API requirements and specifications for the integration.
  - Day 4-5:
    - Develop and test the API endpoints for the integration.
    - Develop and test the authentication and authorization mechanisms for the integration.
  - Day 6-7:
    - Conduct preliminary testing of the integration in a lab environment to evaluate the effectiveness of the integration.
    - Identify any technical issues that need to be addressed.
    - Update the integration plan as needed based on the results of the testing.
- Week 2:
  - Day 1-2:
    - Finalize the API endpoints for the integration.
    - Finalize the authentication and authorization mechanisms for the integration.
    - Conduct comprehensive testing of the integration in a lab environment to identify any issues and bugs.
  - Day 3-4:
    - Conduct real-world testing of the integration with a small group of users in the rural communities.
    - Gather feedback from users and identify any issues that need to be addressed.
  - Day 5:
    - Make necessary updates to the integration based on the feedback and issues identified during testing.
    - Conduct final testing of the integration to ensure stability and reliability.
- Week 3:
  - Day 1-3:
    - Develop and test integration with payment gateways, banks, and other financial institutions that were identified in Week 1.
    - Optimize the integration based on the results of the testing.
  - Day 4-5:
    - Conduct real-world testing of the integration with a larger group of users in the rural communities.
    - Gather feedback from users and identify any issues that need to be addressed.
- Week 4:
  - Day 1-2:
    - Make necessary updates to the integration based on the feedback and issues identified during testing.
    - Conduct final testing of the integration to ensure stability and reliability.
  - Day 3-5:
    - Develop and test any additional features or improvements to the integration that were identified during testing.

4. Develop Prototype

- Week 3:
  - Day 1:
    - Review the research and recommendations from Card 1 and Card 2 to ensure that the proposed solution is feasible and effective.
    - Develop a plan for building the prototype.
  - Day 2-3:
    - Set up the development environment and necessary tools.
    - Develop a prototype of the solution with basic functionality.
    - Document the development process and any technical issues that arise.
  - Day 4-5:
    - Conduct initial testing of the prototype to evaluate its effectiveness.
    - Identify any technical issues that need to be addressed.
    - Update the development plan as needed based on the results of the testing.
- Week 4:
  - Day 1-2:
    - Refine the prototype based on the results of the initial testing.
    - Implement any additional features or improvements identified during testing.
  - Day 3-4:
    - Conduct comprehensive testing of the prototype to identify any issues and bugs.
    - Gather feedback from users and identify any areas for improvement.
  - Day 5:
    - Make necessary updates to the prototype based on the feedback and issues identified during testing.
    - Conduct final testing of the prototype to ensure stability and reliability.
- Week 5:
  - Day 1-2:
    - Begin development of the full solution based on the prototype.
    - Implement any additional features or improvements identified during testing.
  - Day 3-4:
    - Conduct comprehensive testing of the solution to identify any issues and bugs.
    - Gather feedback from users and identify any areas for improvement.
  - Day 5:
    - Make necessary updates to the solution based on the feedback and issues identified during testing.
    - Conduct final testing of the solution to ensure stability and reliability.
- Week 6:
  - Day 1-3:
    - Conduct final testing and debugging of the solution.
    - Prepare the solution for release, including finalizing documentation, creating release notes, and preparing marketing materials.
  - Day 4-5:
    - Release the solution to users and monitor its performance.
    - Address any technical issues or user feedback that arises.

5. Local storage implementation

- Week 1:
  - Day 1:
    - Review the research and recommendations from Card 1 and Card 2 to ensure that the proposed solution is feasible and effective.
    - Develop a plan for building the local storage solution.
  - Day 2-3:
    - Set up the development environment and necessary tools.
    - Develop a basic prototype of the local storage solution.
    - Document the development process and any technical issues that arise.
  - Day 4-5:
    - Conduct initial testing of the local storage solution to evaluate its effectiveness.
    - Identify any technical issues that need to be addressed.
    - Update the development plan as needed based on the results of the testing.
- Week 2:
  - Day 1-4:
    - Refine the local storage solution based on the results of the initial testing.
    - Implement any additional features or improvements identified during testing.
    - Conduct comprehensive testing of the local storage solution to identify any issues and bugs.
    - Gather feedback from users and identify any areas for improvement.
  - Day 5:
    - Make necessary updates to the local storage solution based on the feedback and issues identified during testing.
    - Begin integrating the local storage solution with the online payments platform.
    - Implement any additional features or improvements identified during testing.

6. Unit testing and feedback

- Day 1:
  - write unit tests and run to ensure everything checks out
- Day 2-4:
  - Conduct comprehensive testing of the integration to identify any issues and bugs.
  - Gather feedback from users and identify any areas for improvement.
- Day 5:
  - Make necessary updates to the integration based on the feedback and issues identified during testing.
  - Conduct final testing of the integration to ensure stability and reliability.

b. Task Breakdown

Card 1: Research and Evaluate Solutions

Estimate: 1 week

Summary: Conduct research on available technologies and solutions for improving online payments platform adoption in rural areas. Evaluate the cost, effectiveness, and scalability of each solution. Recommend the best solution for our use case.

Assignee: Senior Engineer

Card 2: Develop Proof of Concept

Estimate: 2 weeks

Summary: Develop a proof of concept for the recommended solution. Test the solution in a staging environment to evaluate its effectiveness and identify any technical issues that need to be addressed.

Assignee: Junior Engineer 1

Card 3: Prototype Development

Estimate: 3 weeks

Summary: Develop a prototype of the solution based on the results of the proof of concept. Test the prototype in a real-world environment to evaluate its effectiveness and identify any issues that need to be addressed.

Assignee: Mid-Level Engineer 2

Card 4: Optimization and Integration

Estimate: 4 weeks

Summary: Optimize the solution for maximum effectiveness and scalability. Integrate the solution with existing systems and infrastructure as needed. Conduct thorough testing to ensure the solution is stable and reliable.

Assignee: Senior Engineer

Card 5: Local Storage Implementation

Estimate: 2 weeks

Summary: Implement local storage (using Room database) for offline usage of the online payments platform on the mobile app. This will allow users to access the platform even when they are offline.

Assignee: Senior Engineer

Card 6: Testing

Estimate: 1 week

Summary: Testing and debugging.

Assignee: Junior Engineer

c. Best Practices and Tools

To ensure the stability of the solution and the mobile app in general, the following best practices and tools will be used:

- Version control system (e.g. Git) for managing code changes and collaborating with the team
- Continuous integration and deployment (CI/CD) tools (e.g. Jenkins, CircleCI) for automating the build, test, and deployment process
- Automated testing tools (e.g. JUnit, Espresso) for testing the app on different devices and configurations
- Code reviews and pair programming to catch bugs and ensure code quality
- Monitoring tools (e.g. Firebase Crashlytics) for identifying and addressing issues in real-time.