AGILE MODEL

PROJECT TITLE: LS LEARNING APP

PREPARED BY: SAM RAJAN

DATE: 23 JULY 2025

SITUATION

- Currently, LS does not have a dedicated mobile app; learners must access courses via desktop websites or offline channels.
- This limits accessibility, as many users prefer convenient, anytime-anywhere mobile learning.
- Demo course videos and live class enrollment occur through separate or manual processes, leading to delays and user inconvenience.
- Payment for course enrollment is not seamlessly integrated, requiring users to follow complex offline or web-based steps.

P R O B L E M

- Absence of a mobile app creates barriers to quick and easy access to learning content and live classes.
- Potential learners lose interest or drop out due to the fragmented experience of browsing demos, registering, and paying via different platforms or channels.
- Lack of mobile optimization results in poor engagement, limited reach, and reduced competitive advantage.
- Manual or non-integrated payment processes reduce convenience and increase chances of errors or abandoned transactions.

O P P O R T U N I T Y

- Developing the LS Learning App will provide a unified, mobile-first platform for users to log in, watch demo videos, and enroll in live classes directly.
- Integrated and secure in-app payment options will streamline the enrollment process, increasing conversions.
- Accessibility anytime and anywhere via mobile devices will expand LS’s audience and improve learner engagement.
- Enhanced user experience through a dedicated app will strengthen brand reputation, encourage repeat usage, and boost customer satisfaction.

PURPOSE STATEMENT (GOALS)

- To develop a user-friendly mobile app that allows customers to easily access demo course videos anywhere, anytime.
- To provide an integrated platform where users can securely make payments to join live classes directly within the app.
- To enhance the overall learning experience by offering seamless navigation and interactive course previews.
- To increase student engagement and enrollment by simplifying the onboarding and payment process and increase the converstion rate .
- To strengthen LS’s brand presence in the mobile education market by delivering a reliable and accessible learning solution.

PROJECT OBJECTIVES

- Design and implement the LS Learning App with features for customer login, demo video access, and secure payment processing.
- Ensure the app is optimized for mobile devices, supporting smooth performance across Android and iOS platforms.
- Develop a secure and seamless in-app payment gateway for live class enrollment.
- Provide an intuitive user interface that enhances user experience and facilitates easy course discovery.
- Integrate backend services to manage course content, user authentication, and payment transactions efficiently.

SUCCESS CRITERIA

- Successful deployment of the mobile app on major app stores with zero critical bugs.
- Minimize system downtime and increase response times fast for preview demo course.
- Payment processing is completed securely with a transaction success rate of 99% or higher.
- The SMART (Specific, Measurable, Achievable, Realistic, and Time- Bound) objectives of the project are

Specific: Requirement

Measurable: Accuracy

Achievable: Feasible

Realistic: Aligned

Time-Bound: Deadline

AGILE METHOD

- Flexibility: Easily accommodate changes such as adding new payment options or enhancing video playback features.
- Early Value Delivery: Users can start accessing core features (login, demo videos) early while other features are developed.
- Stakeholder Collaboration: LS management and end users provide constant feedback, ensuring the product meets real needs.
- Risk Mitigation: Iterative deliveries minimize the risk of late surprises or project failures.

APPROACH

- **Sprint Planning and Backlog Creation**
  - Break down the project into small, manageable user stories (e.g., user login, demo video access, payment gateway integration).
  - Prioritize the product backlog based on business value and technical dependencies.
  - Define sprint goals aligning with high-priority features for each 2-4 week sprint.
- **Roles**
  - Product Owner: Represents LS stakeholders, responsible for defining and prioritizing user stories and acceptance criteria.
  - Scrum Master: Facilitates the Agile process, removes impediments, ensures team adherence to Scrum practices.
  - Development Team: Cross-functional members including developers, testers, UX designers working collaboratively to deliver increments.
- **Sprint Execution and Review**
  - Conduct daily stand-ups to discuss progress, blockages, and plans.
  - Deliver potentially shippable product increments every sprint.
  - Hold sprint reviews to demo features like customer login, demo video streaming, or payment checkout to stakeholders for feedback.
- **Continuous Testing and Integration**
  - Employ continuous integration tools for faster build and test cycles.
  - Automate testing for key functionalities to ensure reliability and performance through sprints.
- **Retrospectives and Adaptation**
  - At the end of each sprint, conduct retrospectives for the team to reflect on what went well, challenges faced, and improvement opportunities.
  - Adapt the process and backlog priorities based on feedback and changing requirements.

R E S O U R C E S :

People:

- Project Manager,Product Owner,Scrum Master ,UI/UX Designer, Developers, QA Engineer(s),DevOps Engineer
- *Team size: 8–11 members*

Time:

- 3–5 months total 2–4 week Agile sprints delivering iterative features

Budget:

- 2 CRORE depending on scope and timeline,Covers payment gateway, cloud services, app store fees, and tools

Other Resources:

- Design & development tools (Figma, IDEs)
- Cloud infrastructure and payment gateway integration
- Testing devices (Android/iOS)
- Collaboration platforms (Jira, Slack, Zoom)
- Security and compliance tools

RISKS

- Technology & Integration Risk: Selecting incompatible or unproven technology stacks, or challenges integrating secure payment gateways and video streaming services, could lead to delays and functional issues.
- Resource and Team Risk: Loss or unavailability of key team members (developers, QA, Scrum Master) or skill gaps can slow development and reduce quality.
- Security & Compliance Risk: Potential vulnerabilities in app security, data protection, and payment processing could lead to data breaches and regulatory non-compliance.
- Stakeholder Communication Risk: Insufficient or unclear communication among stakeholders and the development team may lead to misaligned expectations, scope creep, and rework.

DEPENDENCIES

- Third-Party Services: Dependence on external payment gateways, video hosting platforms, and cloud infrastructure services requires stable APIs and cooperation for smooth integration and performance.
- App Store Approval: The project timeline depends on app store review and approval processes (Apple App Store, Google Play), which can introduce delays or unforeseen compliance requirements.
- Platform Compatibility: Dependence on mobile OS updates and device compatibility necessitates ongoing testing and adjustments, impacting deployment schedules.
- Stakeholder Feedback: Timely input and approvals from LS management and end users during sprint reviews are essential for iterative improvements and avoiding late-stage changes.