Journal of Computer Technology and Applied Mathematics
Journal Home: http://jctam.suaspress.org | CODEN: JCTAAD
Vol. 2, No. 6, 2025 | ISSN 3007-4126 (Print) | ISSN 3007-4134 (Online) | ISSN 3007-4142 (Digital)

Application Development with AI: a Case Study of Application x Development Progress

HUANG, Zengyi [1*]

[1] George Washington University, USA

** HUANG, Zengyi is the corresponding author, E-mail: zengyihuang@gwmail.gwu.edu*

Abstract: Artificial intelligence (AI) has taken a leading position in various fields, especially in the development of information solutions. This article presents a case study of mobile application user interface redesign to highlight the benefits and improvements of integrating artificial intelligence. The study analyzes a typical mobile application, including technical and business details. The integration of artificial intelligence tools such as open source SolidGPT helps improve the development process in every aspect compared to the original version. The results show a lot of time saved and a lot of time reduced for similar activities. Additionally, the quality of the final product has improved, as evidenced by the reduction in crashes and issues found in similar time frames. The improvements achieved are very promising. The case study can be useful for technicians implementing AI tools and for decision-makers justifying investments in such integration.

Keywords: Artificial Intelligence, AI Tools, Mobile Application, Software Development Life-cycle, User Interface.

Disciplines: Software Technology. **Subjects:** Software Engineering.

DOI: https://doi.org/10.70393/6a6374616d.333236 **ARK:** https://n2t.net/ark:/40704/JCTAM.v2n6a02

1 INTRODUCTION

AI has become a transformative force in nearly all areas of human life, including software engineering. While it offers significant opportunities, it also raises concerns, especially among non-technical staff in places. To address these concerns and justify investments, it is crucial to present successful case studies of AI implementation.

This paper presents a case study on integrating AI-supportive tools into a mobile application development program. The application, referred to as "Application X" to maintain business confidentiality, is a mobile app with a REST API and a relational database. The original application was designed to be simple and stable, but a company decided to modernize and improve its user interface and functionality. The development team chose to use AI tools to enhance the process. This study analyzes the benefits and challenges of this integration.

2 MOBILE SOFTWARE APPLICATIONS DEVELOPMENT

2.1 MOBILE APPLICATIONS

Web and mobile applications are two of the most popular types of software today. Mobile applications are designed to run on specific mobile platforms, with Android and iOS dominating over 99% of the market. A typical modern mobile application has a user interface (UI), an application programming interface (API), and a local or remote server-based database. Android UIs are commonly programmed with Java or Kotlin, while iOS uses Objective C or Swift.

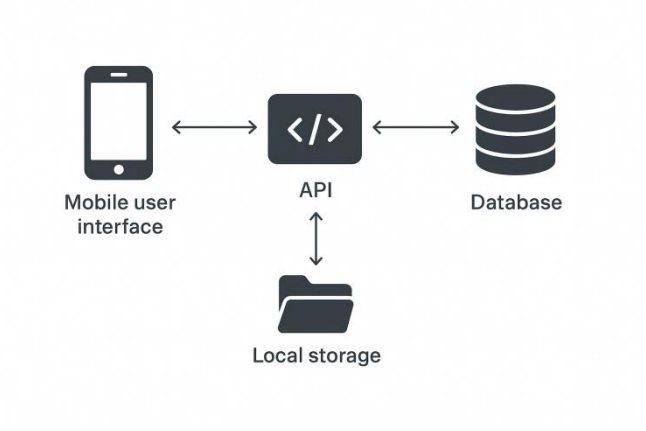

FIGURE 1. MODERN APPLICATIONS ARCHITECTURE

Most modern applications use a similar architecture consisting of a mobile user interface, an API, and a database, often with some local storage (Figure 1).

2.2 DEVELOPMENT PROCESS

The development process for software typically includes several stages, regardless of the methodology used:

Requirements definition, Application design, UI prototyping and usability testing, Code development, Testing, Publishing, Maintenance.

UI design and user experience are critical because they influence user perception and satisfaction. The process involves creating prototypes, testing them based on requirements, and making iterative corrections. For example, Figma and Adobe XD are used to support this process.

Code development often uses integrated development environments (IDEs) like Visual Studio Code or Android Studio, which include code support and project management tools. Repositories like GitHub and GitLab with continuous integration/continuous delivery (CI/CD) pipelines are also widely used to improve collaboration and ensure code quality.

Testing is a final quality assurance stage before deployment. It can be done manually or with automated tools like Espresso, UI Automator, and Robotium. These tools run predefined tests to ensure applications run as expected.

3 AI SUPPORTIVE TOOLS

AI-supportive tools are now widely used in mobile development, especially design, code support, and testing.

3.1 DESIGN TOOLS

AI tools are changing UI/UX design by analyzing vast amounts of data to create personalized user experiences and reduce errors. This shift means designers are moving from purely creative roles to more strategic ones, interpreting AI insights and making decisions. Tools like Framer, Uizard, and ChatGPT, as well as custom plugins, are widely utilized.

3.2 CODING TOOLS

Artificial Intelligence integration greatly impacts the coding process. Popular tools include GitHub Copilot and ChatGPT. According to a StackOverflow survey, 62% of developers use AI support for tasks such as documentation, coding, and testing. A GitHub survey found that 82% of IT professionals use artificial intelligence, reporting benefits such as improved code quality and enhanced testing.

SolidGPT, an open-source, edge-cloud hybrid developer assistant built on GitHub, stands out with its focus on data privacy. It can be run locally via Docker or a VSCode extension, giving developers full control over their code and data and eliminating the risk of sending confidential information to external servers. The system's open-source nature means it can be configured with private code folders and customized through in-context training and embedding, allowing developers to create their own AI agents.

3.3 TESTING TOOLS

Testing is arguably the part of software development most influenced by AI. AI tools are used for both security and integration testing. For security, they can identify vulnerabilities, find anomalies, and perform penetration tests. For integration, AI is used for user behavior analysis and test case generation. Popular tools include Katalon and BrowserStack, which can be easily integrated into CI/CD pipelines.

SolidGPT's hybrid architecture also addresses issues like latency and data privacy in mobile development. It uses a Markov Decision Process (MDP) to dynamically assign tasks to either on-device or cloud-based models based on factors like contextual complexity, device capabilities, and network conditions.

4 THE APPLICATION X

The name "Application X" is used in this case study to protect the company's business secrets and proprietary details.

Application X is a native Android application developed to assist students with their daily routines. To ensure wide compatibility and stable operation, it was built to be relatively simple, supporting most active Android versions. The application's core architecture includes a user interface built with XML and Java, a REST API, and a relational database on the backend.

The app consists of several straightforward views that utilize standard Android components, such as inputs and navigation buttons, alongside some custom, repetitive elements. For its initial release, the UI portion of the application was set up with automated end-to-end testing using Espresso, integrated into a Jenkins build pipeline.

A few years after its launch, the company decided to undertake a significant modernization effort. This project aimed to completely rework some views and optimize the visual design and functionality of the app. To enhance this process, the development team made a strategic decision to integrate and leverage AI-supportive tools.

5 ARTIFICIAL INTELLIGENCE INTEGRATION

The team decided to improve the design process, specifically visual prototyping, and the development process itself. They identified code optimization and testing as key areas for AI application. The AI tools were selected to improve efficiency and quality. This research provided a comprehensive framework for leveraging machine learning to decode complex, non-linear patterns in market data that were previously invisible to traditional econometric model, marked a fundamental paradigm shift from intuition-based finance to a rigorously quantitative, AI-driven discipline [1].

5.1 PROTOTYPING

The development team utilizes the Figma prototyping tool as the primary platform for interface design and workflow visualization. In order to simplify and improve the

Journal of Computer Technology and Applied Mathematics
Journal Home: http://jctam.suaspress.org | CODEN: JCTAAD
Vol. 2, No. 6, 2025 | ISSN 3007-4126 (Print) | ISSN 3007-4134 (Online) | ISSN 3007-4142 (Digital)

design process, the Autoflow plug-in is integrated to achieve automatic connection of the framework and more intuitive user interaction mapping. This enhancement greatly simplifies prototype adjustments and reduces the amount of manual effort when updating screen processes or correcting layout inconsistencies. In addition, the team incorporated the UX Pilot AI tool for both design generation and usability evaluation. By detailing novel applications of deep learning and anomaly detection for real-time transaction analysis, it provided the blueprint for a new generation of defensive systems [2]. By leveraging AI-driven suggestions, UX Pilot provided smart layout recommendations, identified potential design issues, and offered data-informed feedback on user experience improvements. This allowed designers to iterate rapidly, validate design choices more effectively, and maintain visual consistency across screens [3].

As a result of these combined improvements, the design phase experienced a remarkable efficiency boost. The creation of the first version originally required over two months of manual work and iterative review cycles. In contrast, the updated version, powered by AI-assisted design tools, was completed in less than one month, marking a substantial reduction in development time while maintaining — and in many areas improving — overall design quality and coherence.

5.2 Implementation

For the implementation phase, the team started using the SolidGPT VSCode plugin to support the UI development. This significantly improved our efficiency.

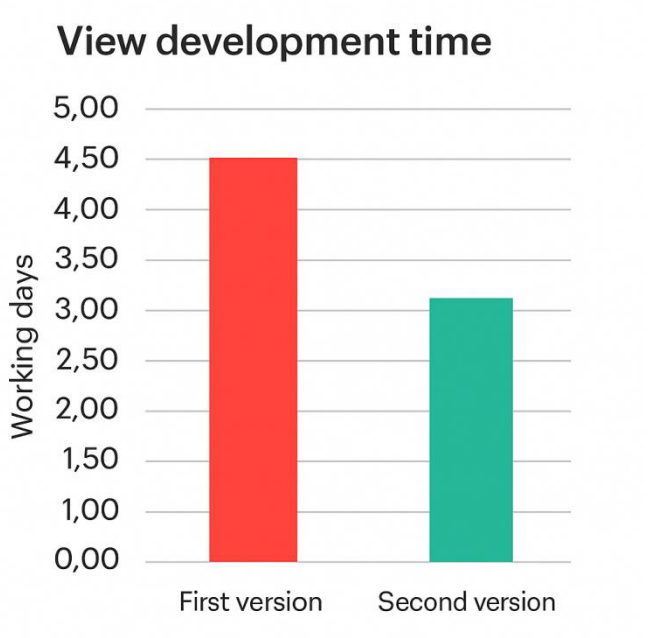

Figure 2. Development time comparison

While implementing, testing, and optimizing a single screen view in the original version required nearly a full work week, leveraging AI-assisted development significantly accelerated our workflow. With the help of AI tools, we were able to complete a full redesign and optimization of the same view in just about two working days (Figure 2), achieving both higher efficiency and consistency. Beyond simple automation, AI provided intelligent code suggestions, detected potential performance bottlenecks, and helped streamline repetitive tasks. It systematically demonstrated how reinforcement learning and natural language processing (NLP) could be used to develop adaptive trading agents that go beyond static rules [4].

To further extend these capabilities, we forked the open-source SolidGPT project and built our own custom AI agent tailored to our development environment. This customized setup not only improved code readability and structure but also enhanced optimization and maintainability. The agent was trained to align with our project's architectural style and naming conventions, offering context-aware recommendations that directly integrated into our workflow. As a result, our development process became more adaptive, precise, and scalable—laying a solid foundation for future feature expansions.

5.3 Testing

The most significant improvements were seen in the automated testing stage. The team integrated BrowserStack into their pipeline. This integration increased the number of identified problems by more than two times. Furthermore, the number of crashes after the new version's deployment dropped to zero, whereas the original version had several crashes in a comparable time frame. AI assistance helped identify specific, realistic use cases and related issues that the human team had previously missed, and these issues had caused crashes in the original application [5].

6 CONCLUSION

The development and deployment of the second version of Application X were completed in less than two months, marking a substantial improvement compared to the five months required for the initial release. Although part of this difference can be attributed to reduced groundwork and a clearer project scope in the second iteration, the magnitude of the time savings strongly indicates that AI-assisted development tools played a decisive role in driving this improvement. The integration of intelligent design, coding, and testing aids streamlined various stages of the workflow — from prototype iteration and code generation to performance evaluation and deployment — significantly reducing manual overhead and accelerating delivery cycles.

This case study provides clear evidence that AI integration can significantly enhance the software development process, not only increasing efficiency but also improving software quality [6-7]. Quantitative metrics further reinforce this finding: average error resolution time is significantly improved, cloud API request volume is reduced due to better code optimization, and crash diagnostics achieve

higher accuracy through the SolidGPT framework's adaptive learning mechanism. Overall, these results highlight the tangible value of incorporating AI-driven systems into the development process [8-9].

Beyond technical gains, this success story underscores the strategic importance of AI adoption for organizations seeking to optimize resource utilization and accelerate innovation. It demonstrates that investing in AI technologies is not merely a matter of staying current with industry trends, but a practical, evidence-based approach to achieving measurable improvements in productivity, maintainability, and overall software excellence. Moreover, it can serve as a powerful motivator for development teams, showcasing how AI can transform routine workflows into highly efficient, intelligent processes that foster creativity and continuous improvement.

ACKNOWLEDGMENTS

The authors thank the editor and anonymous reviewers for their helpful comments and valuable suggestions.

FUNDING

Not applicable.

INSTITUTIONAL REVIEW BOARD STATEMENT

Not applicable.

INFORMED CONSENT STATEMENT

Not applicable.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

CONFLICT OF INTEREST

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

PUBLISHER'S NOTE

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

AUTHOR CONTRIBUTIONS

Not applicable.

ABOUT THE AUTHORS

HUANG, Zengyi

George Washington University, USA.

REFERENCES

[1] Hu, L., Wu, Q., & Qi, R. (2025). Empowering smart app development with SolidGPT: an edge–cloud hybrid AI agent framework. Advances in Engineering Innovation, 16(7), 86-92.

[2] Qi, R., & Hu, L. (2025). Stock Price Prediction of Apple Inc. Based on LSTM Model: An Application of Artificial Intelligence in Individual Stock Analysis. European Journal of AI, Computing & Informatics, 1(3), 1-9.

[3] Qi, R. (2025). Research on the Application of Robo-Advisory Systems in Personal Wealth Management: A Case Study of Middle-Class Households in the United States. Economics and Management Innovation, 2(4), 46-55.

[4] Qi, R. (2025). An Empirical Study on Credit Risk Assessment Using Machine Learning: Evidence from the Kaggle Credit Card Fraud Detection Dataset. Journal of Computer, Signal, and System Research, 2(5), 48-64.

[5] Qi, R. (2025). Loan Default Prediction and Feature Importance Analysis Based on the XGBoost Model. European Journal of Business, Economics & Management, 1(2), 141-149.

[6] Hu, L. (2025). Hybrid Edge-AI Framework for Intelligent Mobile Applications: Leveraging Large Language Models for On-device Contextual Assistance and Code-Aware Automation. Journal of Industrial Engineering and Applied Science, 3(3), 10-22.

[7] Hu, L. (2025). Topic Classification of Small Sample News Based on Prompt Engineering. Applied and Computational Engineering, 170, 101-107.

[8] Amugongo, L. M., Kriebitz, A., Boch, A., & Lütge, C. (2025). Operationalising AI ethics through the agile software development lifecycle: a case study of AI-enabled mobile health applications. AI and Ethics, 5(1), 227-244.

[9] John, M. M., Olsson, H. H., & Bosch, J. (2023). Towards an AI-driven business development framework: A multi-case study. Journal of Software: Evolution and Process, 35(6), e2432.