FMDB *Transactions on Sustainable Computing Systems*

Development and Testing of Incorporated ASM with MVP Architecture Model for Android Mobile App Development

N. Rajasekaran[1, *], S. M. Jagatheesan[2], S. Krithika[3], Julio Suárez Albanchez[4]

[1]Department of Computer Applications, Kongu Arts and Science College (Autonomous), Erode, Tamil Nadu, India.
[2]Department of Computer Science, Gobi Arts and Science College, Gobi, Tamil Nadu, India.
[3]Department of Computer Science (PG), Kongu Arts and Science College (Autonomous), Erode, Tamil Nadu, India.
[4]Department of Higher School of Engineering and Technology, International University of La Rioja, La Rioja, Spain.
rajasekarandpm@gmail.com[1], smjagatheesan@gmail.com[2], krithitup86@gmail.com[3], julio.suarez@unir.net[4],

Abstract: Successful mobile application development with Android is done by using different software development architecture models. Model View Presenter (MVP) is the most popular and preferable architecture for Android app developers. This type of architecture gives modularity to the software projects and ensures that all the codes in different modules are easily covered in Unit testing. It makes the development and testing activities very easy for the developers to maintain, enhance, and expand the different features of the particular application. We proposed Agile Scrum model-based MVP architecture (ASM-MVP) for the Android application development life cycle. Agile Scrum is the most popular development model, which gives continuous integration, fast delivery, and customer collaboration in each software development sprint. So, we integrate the Agile Scrum process with MVP architecture for our proposed app development process, and it's validated through performance testing using JMeter. We assign the same project to be built by different agile teams. Out of five agile project teams, one team should follow our proposed architecture model, and the other three teams can use their strategy for development. The development process is extended to five different mobile app projects. The developed app performance was tested through Jmeter, and the results were compared. The result shows that the project developed using our proposed model gives better results than others.

Keywords: Development and Testing; Android Mobile App Development; Model View Presenter (MVP); Plan-Driven Processes; Agile Methods; Testing Using Jmeter; Agile Scrum Model.

Received on: 05/12/2022, **Revised on:** 07/02/2023, **Accepted on:** 15/03/2023, **Published on:** 13/04/2023

Cited by: N. Rajasekaran, S. M. Jagatheesan, S. Krithika, and J. S. Albanchez, "Development and Testing of Incorporated ASM with MVP Architecture Model for Android Mobile App Development," *FMDB Transactions on Sustainable Computing Systems*., vol. 1, no. 2, pp. 65–76, 2023.

1. Introduction

Music The process of creating mobile applications is known as mobile application development. The mobile applications are made specifically for portable computing platforms like tablets, smartphones, and Personal Digital Assistants. Due to the constant changes [1] in requirements, the domination of new technologies, the technical limitations of mobile systems, and the high expectations and demands of users, developing mobile applications is currently a very difficult task for app developers. Currently, no models or processes are suitable for creating mobile applications. As a result, the developers are forced to create an application with several problems [2].

*Corresponding author.

We proposed a new ASM-MVP model to address these problems and difficulties. In this model, we use the Model View Presenter paradigm in an Agile-based app development process. The agile methodology and MVP are key components of this strategy for developing mobile applications.

1.1. Agile Methodology

Agile [3] is a philosophy or attitude employed in software development approaches based on incremental and iterative approaches, not software or a programming language. A collection of independent software development process thinkers called "The Agile Alliance" established the agile methodology. They are given a Manifesto [3] that contains 12 key guidelines for agile software development.

Plan-driven processes or agile methods can be used to design mobile applications [4]. Describes a plan-driven process in which all process operations are preplanned and progress is gauged using this plan. Plan-driven is static and not iterative in comparison to agile. It is not appropriate for creating mobile applications. The current environment necessitates the development of agile-based mobile applications since the market is competitive, products must be delivered quickly, development costs must be reduced, and development speed must be increased [5]. Many agile techniques [7] are in use today, but the scrum [6] methodology is the best option for developers and is most frequently employed in the IT sector. The Agile Scrum Model is incorporated with MVP (Incorporated ASM-MVP) in this study, and the outcomes were contrasted with MVC and MVP.

1.2. Architectural Pattern of MVC

The most well-known software design patterns, MVC and MVP, are used primarily to divide applications while processing, visualizing, and managing data for User Interface (UI) applications. The primary objective of this architecture is to improve the application's modularity, flexibility, testability, and maintainability. MVC design pattern splits an application into three main components: Model, View, and Controller. The model can manage and store data and show results to the view in response to user requests made through the controller. Data can be altered and transformed depending on the business logic and rules. The MVC Architecture Patterns are depicted in Figure 1.

The view represents UI elements such as HTML, DHTML, and XML. View is in charge of showing the user the data via the screen after receiving it from the controller. Through the controller, model and view can communicate with one another. The controller takes over incoming requests from the view, processes them through the model, and returns the results to the view. Typically, it serves as a go-between for the View and the Model. Due to simultaneous input from several users, the controller and view are closely related.

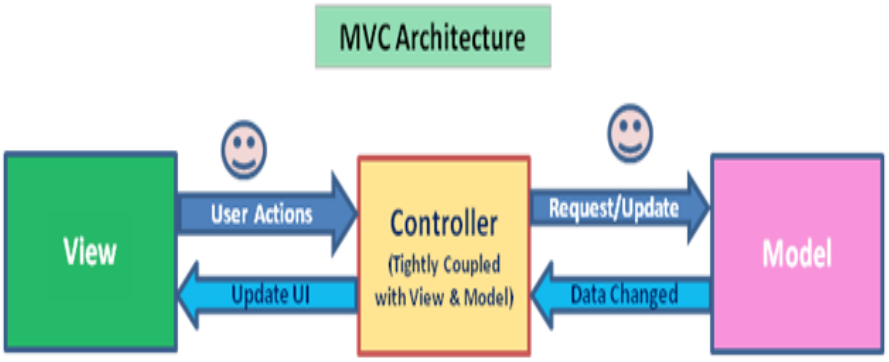

Figure 1: MVC Architecture Pattern

1.3. Architectural Pattern of MVP

The MVP architecture depicted in Figure 2 demonstrates the MVC-like MVP Architecture paradigm. It is derived from MVC; in this instance, the presenter takes the controller role. Model, View, and Presenter are the three main components divided into this design for an application. The presenter plays an impartment role, and the model and view are identical to MVPs. The presenter gets the input or request from the users through the view, processes it with the aid of the model, and then sends the finished product back to the view. The presenter uses an Interface to talk to the viewer.

The presenter class defines an interface to which the necessary data is passed. Additionally, detached from the view, the presenter communicates with it via an Interface. The main benefits of MVP over MVC include easier unit testing due to maximum testability surface, clear separation of the View and Model, less code due to data binding, and loose coupling between the View and Presenter, which communicate with one another via an interface.

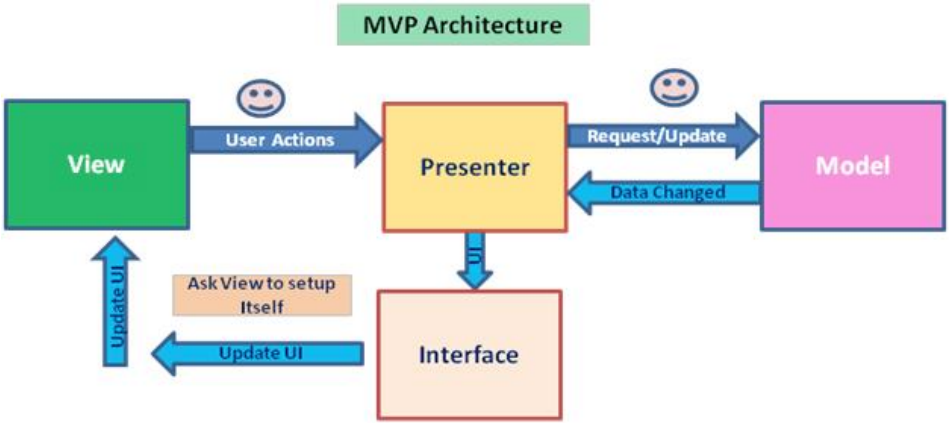

Figure 2: MVP Architecture Pattern

1.4. Incorporated Agile Scrum with MVP (ASM-MVP)

The Agile Scrum Model depicts the ASM-MVP Architecture Pattern in Figure 3. It combines the scrum framework with the agile ideology. The system's objective is to provide stakeholders with the most value possible. It is simple and sprint-based [8]. Agile refers to "incremental" development, which enables software teams to create tasks in manageable chunks. The Scrum model is intended to divide projects into manageable software chunks, or "sprints." Agile scrum methodology is advantageous for companies that must complete particular projects fast.

This approach is an incremental development-based project management solution. Sprints lasting two to four weeks make up each iteration. Building the most crucial features first is the aim of each sprint to produce a potentially marketable product. The product is expanded in succeeding sprints, and adjustments are made in response to stakeholder and consumer feedback in between sprints.

ASM is included in MVP in the suggested model. For a software product that is well-equipped and devoid of errors, we can use the MVP architecture pattern during the creation of each sprint.

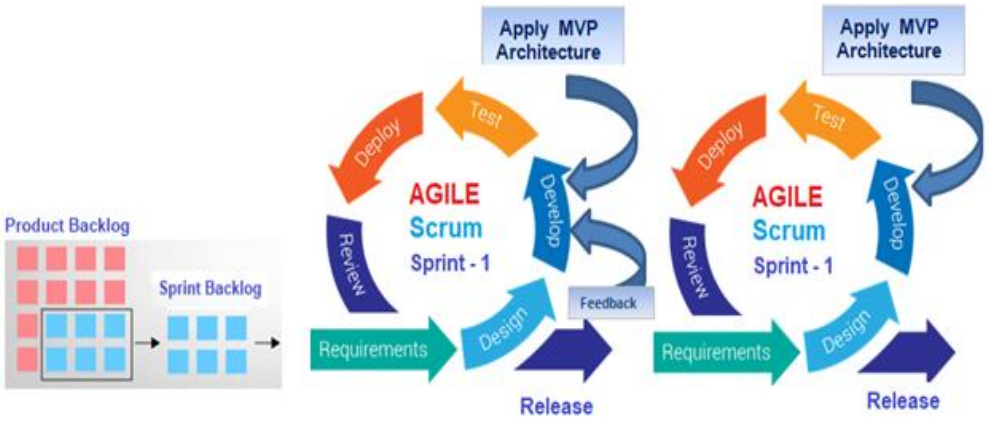

Figure 3: Incorporated ASM-MVP Architecture Pattern

2. Related Work

According to the survey [9], MVC has been the most frequently used design for several years. The most common pattern utilized in the development of mobile applications is this one. MVP is a less common approach used in app development than MVC. By comparing memory use, CPU utilization, and execution time, a study [10] examined the significance and impact of adopting MVP and MVVM (Model View and View Model) architectures on the performance of Android applications. According to the experimental findings, the MVP is superior to the other models in several ways. A technique that derives the MVP design pattern into Modern Driven Architecture (MDA) was presented in this study [11]. The MVP was designed with the aid of MDA and UML.

This study [12] investigated Scrum, Kanban, and JIRA. According to the report, JIRA is essential for tracking project development activities throughout each sprint under the Scrum paradigm, and Kanban is not ideal for developing Android apps. [13] In order to look into the code and testing operations, this study evaluated the quality of the MVVM pattern's existing architectural implementation, paying particular attention to its modifiability, testability, and performance. The MVC architectural pattern divides an application's functions from one another. However, it demonstrates how user interactions are transformed into useful behavior. The MVP architecture enables developers to individually handle user interface and functional behaviors.

The outcome demonstrates that in contrast to MVC, MVP unit testing is quite simple. The modularity and maintainability of two apps created using the MVP design pattern and the anti-pattern were contrasted in this article. By comparing the "anti-pattern" twice, this study demonstrated empirically that using the MVP Design pattern considerably increases modularity.

3. Methodology

1. Select five different Android projects for development and testing.
2. Assign each project to four student groups for development.
   P1→ G1*****, G2, G3 and G4
   P2→ G1, G2*****, G3 and G4
   P3→ G1, G2, G3***** and G4
   P4→ G1, G2, G3 and G4*****
   P5→ G1*****, G2, G3 and G4*****
   ***** →Adopted Proposed Agile Scrum-based MVP Architecture for Development
   Other Groups → choosing their development strategy.
3. At least one group should follow our proposed ASM-MVP Model for development.
4. After the development of each sprint, functional testing will be done.
5. After the functional test, the JMeter tool will measure the application's performance.
6. Result comparison of all the projects.
7. Conclusion

Figure 4 is given below as the Process flow of the proposed work:

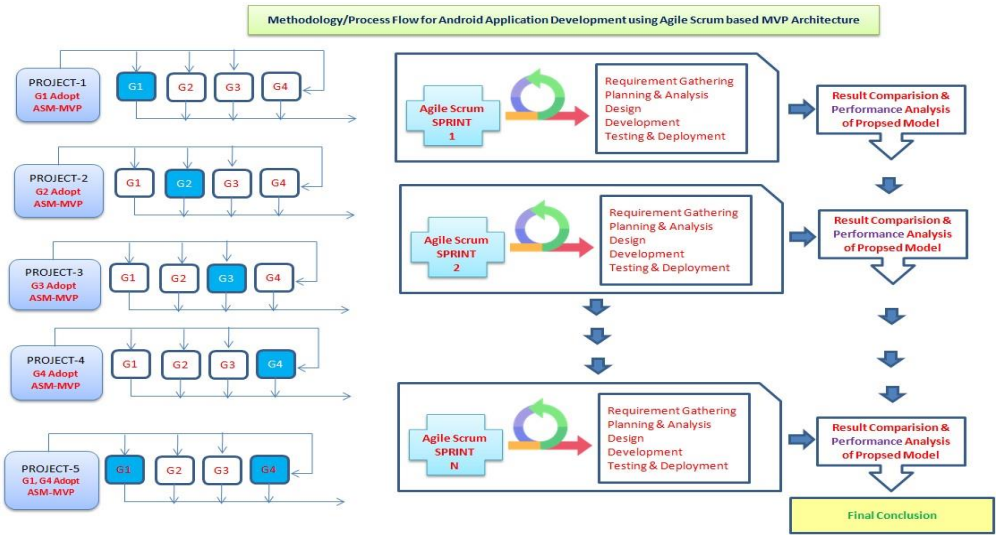

Figure 4: Process flow of the proposed work

4. Implementation

During their internship, the proposed ASM-MVP model was implemented in live projects through the student groups of Kongu Arts and Science College, Erode. Four student groups were created, and each group was assigned the suggested model. Five live projects were also assigned through the company (Fig.5).

Suggested Model	ASM-MVP Suggested to Various Student Groups of KASC for APP Development			
Stu. Groups	G1	G2	G3	G4
Departments	BCA	BCA	B.Sc(CS)	B.Sc(IT)
Internship Providers	C-Cube technologies, Erode.	Angler Technologies, Coimbatore.	TekAfforde Solutions, Bangalore.	Nutz Technovation Pvt Ltd, Erode.
1	Bhanu prasath m	Sriram sengotuvel t	Rohitha M	Hariharan c
2	Manikandan k	Suganesh kumaran r	Dharshini V	Jagadesh s
3	Sujith k	Vimal k	Kishore Raju B	Shashwath p
4	Indran s	Lokesh m	Kishore R	Karthikeyan s
5	Tharun r.S	Arun bhagavathi t	Thrishna M	Kathirvel k

Figure 5: Student groups for implementation

Each project was developed through all groups separately. One group should adopt our ASM-MVP model during the development process, and others can follow any methodology. Likewise, each project was implemented, and the results were compared.

The project assignment is shown in the Figure. 6. Five different Android live projects were assigned to all groups for development. Each group was developed separately. In one project, at least one group should follow our proposed model. In P1, Group 1 follows; in P2, Group 2 follows; in P3, Group 3 follows; in P4, Group 4 follows; in P5, Group 4 and Group 1 follows our proposed ASM-MVP Model for app development.

LIVE Android PROJECTS (P)	G1	G2	G3	G4
	BCA	BCA	B.Sc(CS)	B.Sc(IT)
(P1) students Information System	ASM-MVP	*	*	*
(P2) Placement Management System	*	ASM-MVP	*	*
(P3) Event Management	*	*	ASM-MVP	*
(P4) Online Test System	*	*	*	ASM-MVP
(P5) Shopping Cart Application	ASM-MVP			ASM-MVP

Figure 6: Allocated projects for different groups

All the projects were implemented in Java with Android Studio. Some groups of students developed the projects with MVC architecture; some of them followed MVP, and some of them followed ASM-MVP. The sample coding is shown in Figures 7 to 11.

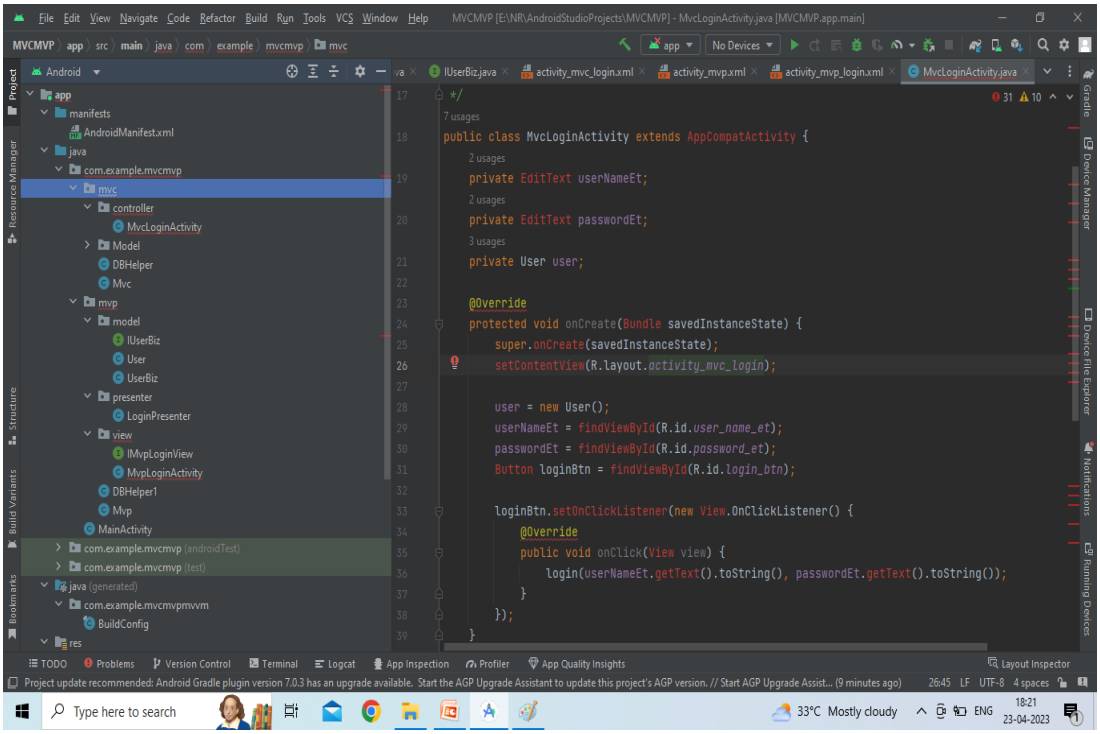

Figure 7: MVC Implementation in Android

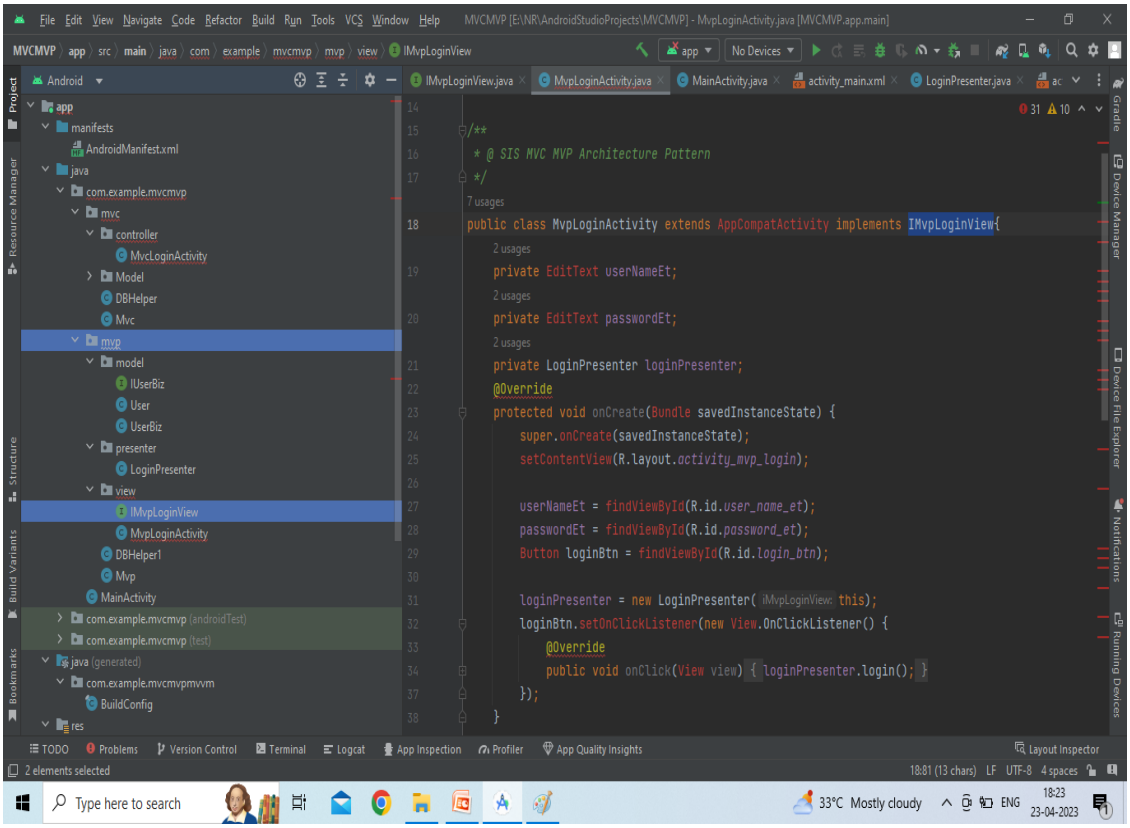

Figure 8: MVP Implementation in Android

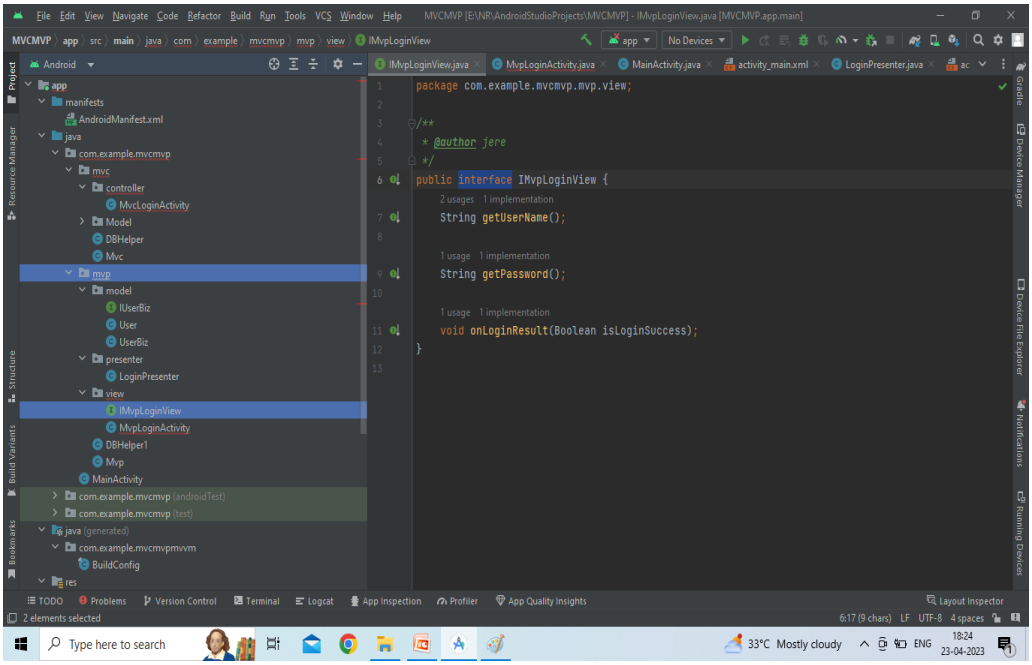

Figure 9: ASM-MVP Implementation in Android

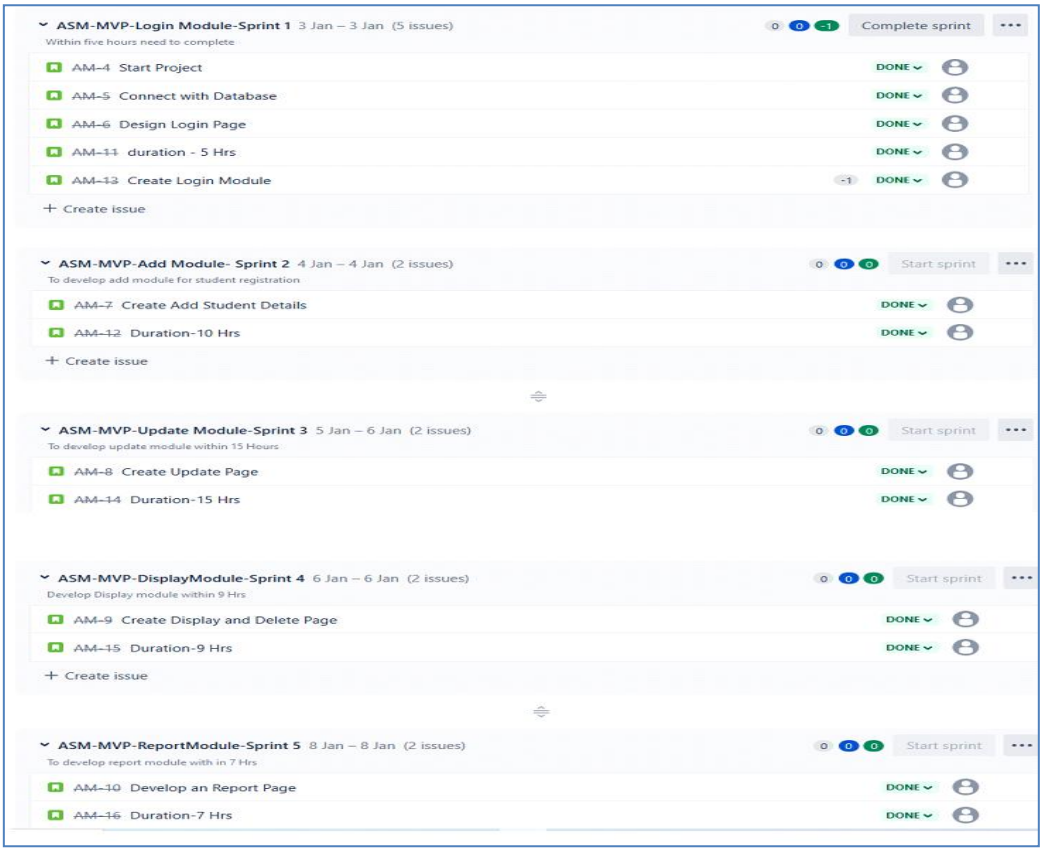

Figure 10: Scrum sprint tracking with JIRA

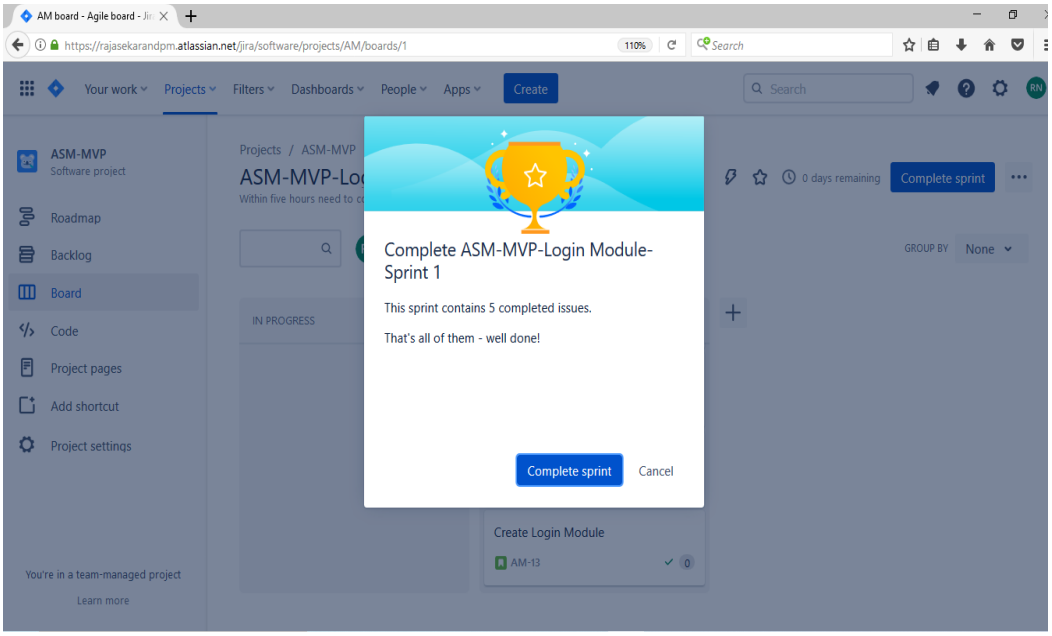

Figure 11: Scrum sprint completed with JIRA

5. Testing and Results

All the projects were developed sprint by sprint. Each sprit was carried out through the JIRA Tool. After completing all the projects, the developers did unit testing. Then, the overall system was tested. After unit and system testing, the non-functional testing was carried out through the JMeter tool. SLOC, Load Time, Connect Time, and Latency were tested during the performance testing (Figures 12 to 17).

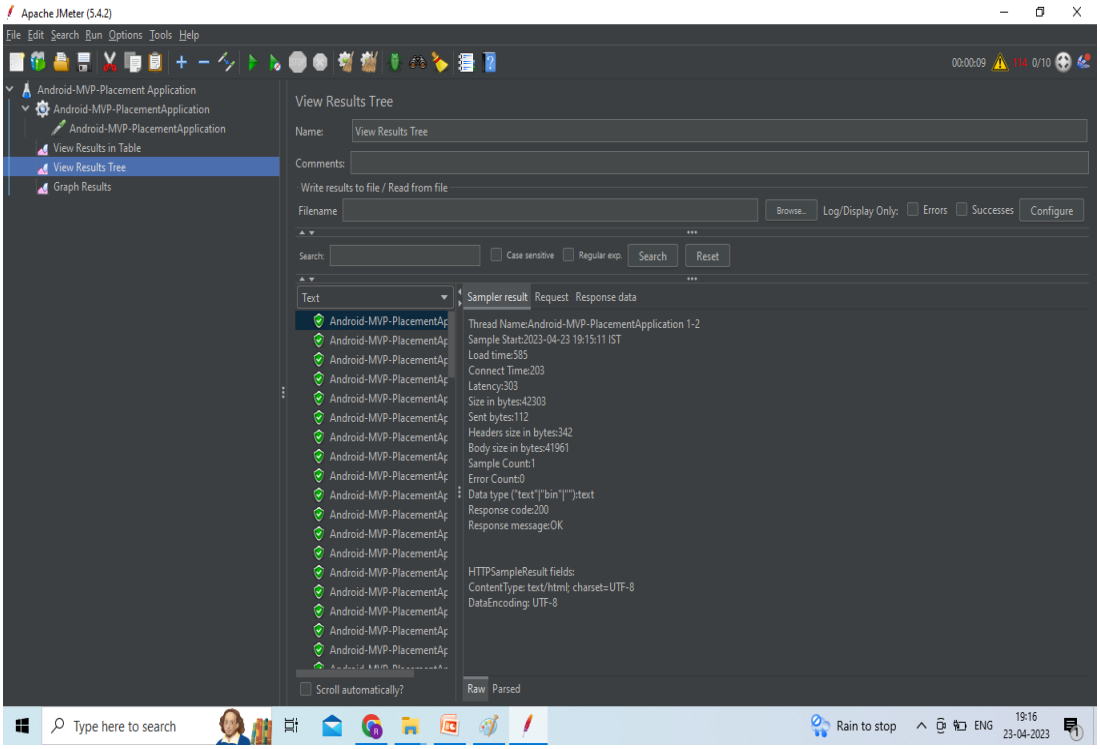

Figure 12: Performance testing with JMeter

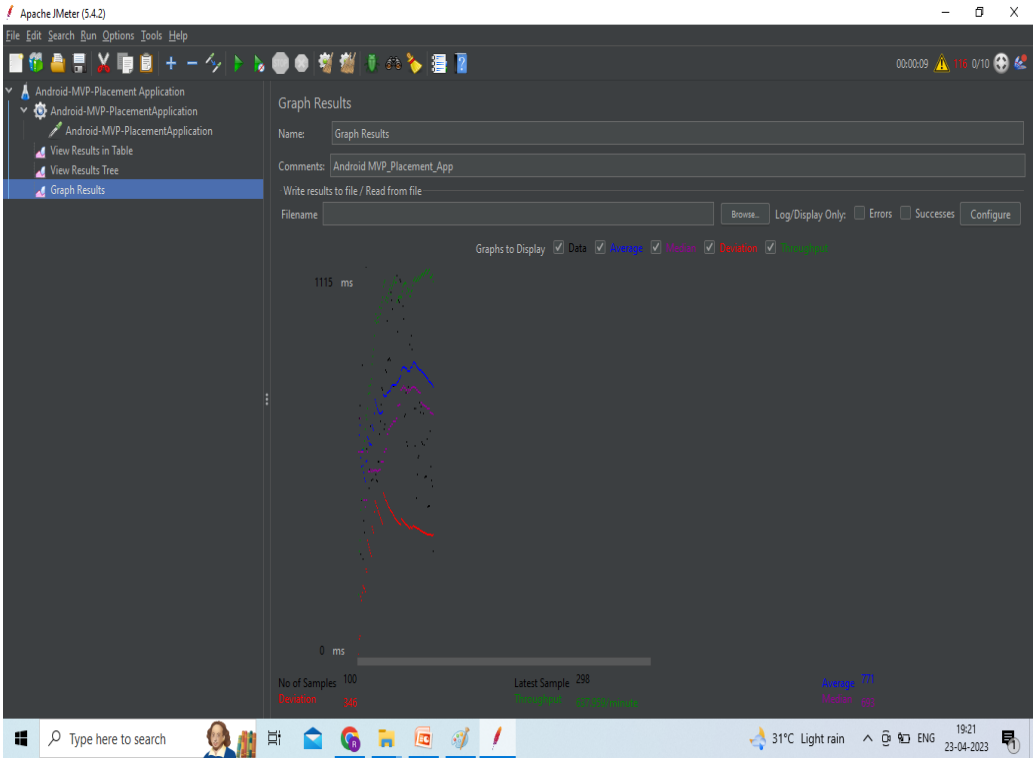

Figure 13: Performance testing with JMeter

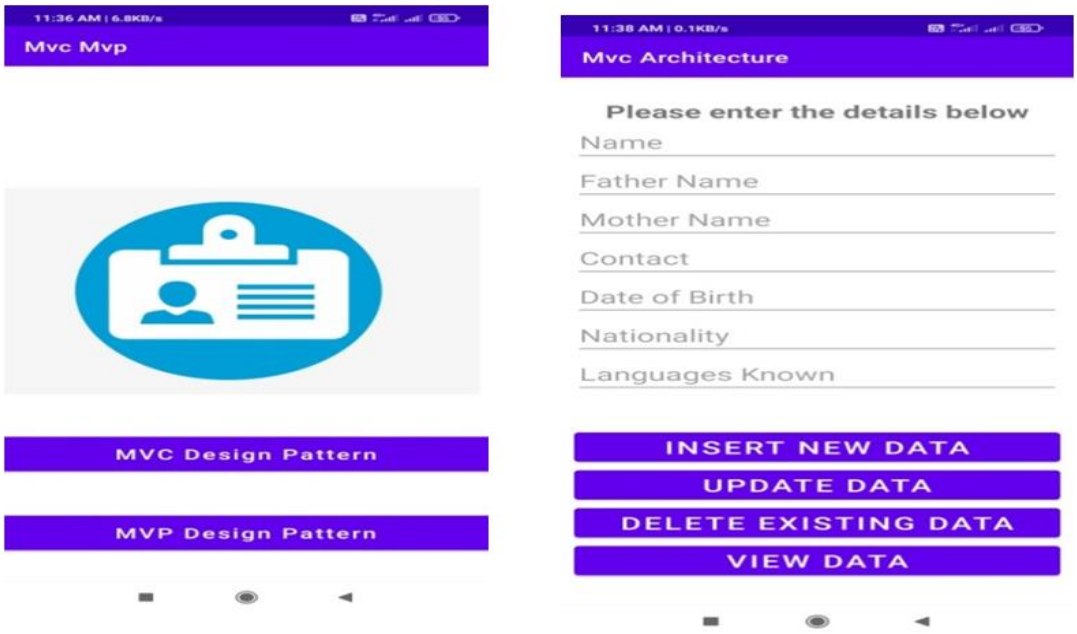

Figure 14: Android App for Student Information System

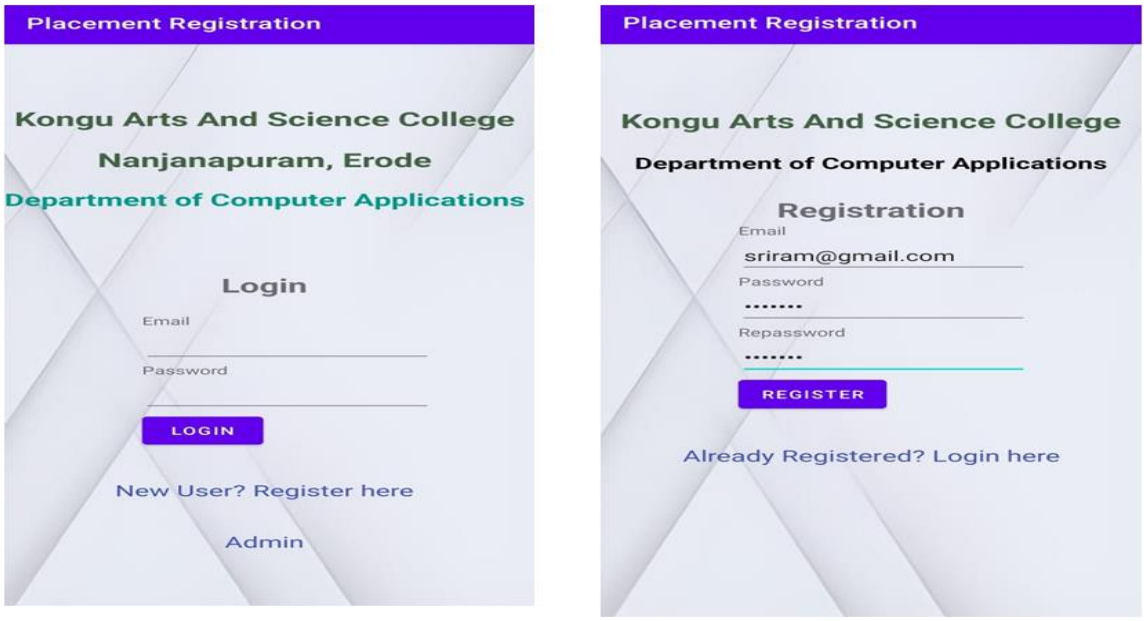

Figure 15: Android App for Placement

Project	Performance	G1	G2	G3	G4	G5
(P1) SIS	*Total Sprints*	***10***	*12*	*10*	*11*	*10*
	SLOC	***1512***	*1410*	*1310*	*1400*	*1380*
	Load Time in Mi. Seconds	***585***	*710*	*600*	*613*	*700*
	ConnectTime in Mi. Seconds	***203***	*203*	*234*	*250*	*300*
	Latency	***303***	*315*	*349*	*400*	*476*
(P2) Placement	*Total Sprints*	*16*	***16***	*15*	*16*	*17*
	SLOC	*1533*	***1612***	*1504*	*1692*	*1655*
	Load Time	*700*	***600***	*690*	*784*	*900*
	Connect Time	*324*	***210***	*330*	*412*	*523*
	Latency	*300*	***250***	*297*	*423*	*495*

Figure 16: Results of P1 and P2

Project	Performance	G1	G2	G3	G4	G5
P3 Event	Total Sprints	15	15	15	16	17
	SLOC	1500	1512	1613	1692	1712
	Load Time	592	600	640	784	914
	Connect Time	300	278	312	400	501
	Latency	250	300	267	423	515
P4 Test Mnt.	Total Sprints	20	20	19	20	18
	SLOC	1650	1700	1525	1674	1504
	Load Time	612	723	600	593	600
	Connect Time	217	250	235	198	300
	Latency	412	390	350	312	419
P5 E-Com	Total Sprints	11	10	9	10	10
	SLOC	1110	1250	1190	1250	1300
	Load Time	692	592	720	430	440
	Connect Time	114	60	102	100	0
	Latency	376	392	289	389	290

Figure 17: Results of P3, P4 and P5

The Proposed Model ASM-MVP, suggested to students' groups, performs better in 3 (P1, P2, P4) Projects in three groups out of 5. P5-Group5 gives almost the same performance as P5-Group4. In P3-Group3 alone, our model gives a little deviation in the performance. Hence, the proposed model performs better in 3 projects with three groups out of five.

6. Conclusion

When designing an application for a small or a large size android project, the Software development architecture pattern becomes very significant. The architecture is a very crucial factor in the process of developing applications for Android specifically. Because of the rapid advancement of technology, MVC and MVP architecture have become viable options for consideration by Android application developers. According to the results of this research, The Proposed Agile Scrum-based MVP Architecture Model was designed and introduced for Android App Development. The ASM-MVP model gives better consequences. The model was validated through the Android live projects with students through their internship period of the project development. Five live projects were allotted to each group for development. The performance of the model was measured effectively through the JMeter. Out of five projects and five groups, it gives better results in three projects (P1, P2, and P4). In P5, it is almost the same in other groups. The ASM-MVP model deviated in only one group (G3). Hence, The ASM-MVP Model gives better results and is almost a suitable choice for Android Developers.

Acknowledgment: The support of all my co-authors is highly appreciated.

Data Availability Statement: This study uses online benchmark data to conduct the research. This is a fresh study done by the authors.

Funding Statement: No funding has been obtained to help prepare this manuscript and research work.

Conflicts of Interest Statement: No conflicts of interest have been declared by the author(s). This is the authors' fresh work. Citations and references are mentioned as per the used information.

Ethics and Consent Statement: The consent has been obtained from the colleges during data collection and has received ethical approval and participant consent.

References

1. Z. Mushtaq, “Mobile Application Development: Issues and Challenges,” International Research Journal of Engineering and Technology, vol. 3, no.8, pp. 1096–1099, 2016.
2. A. Ahmad, K. Li, C. Feng, S. M. Asim, A. Yousif, and S. Ge, “An empirical study of investigating mobile applications development challenges,” IEEE Access, vol. 6, pp. 17711–17728, 2018.
3. A. Alliance, “Agile Software Development Manifesto. Retrieved from Manifesto for Agile Software Development,” Agile Alliance, 2001. [Online]. Available: http://agilemanifesto.org/. [Accessed 20 January 2022].
4. I.Sommerville, “Requirements Engineering,” in Software Engineering, 10th Edition, Boston, Massachusetts, Pearson Education, Inc, USA, pp. 82-84, 2016.
5. M. M. Kirmani, “Agile Development Method for Mobile Applications: A Study,” International Journal of Advanced Research in Computer Science, vol. 8, no. 5, 2017.
6. L. Ghandi, C. Silva, D. Martinez and T. Gualotuña, "Mobile application development process: A practical experience," 2017 12th Iberian Conference on Information Systems and Technologies (CISTI), Lisbon, Portugal, pp. 1-6, 2017. doi: 10.23919/CISTI.2017.7975825.
7. M. Wadhwa and N. Sharma, “Review of agile software development methodologies,” Advances in Computer Science and Information Technology, vol. 2, no. 4, pp. 370–374, 2015.
8. H. K.Flora, S. V. Chande, and X. Wang, “Adopting an agile approach for the development of mobile applications,” Int. J. Comput. Appl., vol. 94, no. 17, pp. 43–50, 2014.
9. A. Daoudi, G. ElBoussaidi, N. Moha, and S. Kpodjedo, “An exploratory study of MVC-based architectural patterns in Android apps,” in Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, USA, pp. 1711–1720, 2019.
10. B. Wisnuadhi, G. Munawar, and U. Wahyu, “Performance comparison of native android application on MVP and MVVM,” in Proceedings of the International Seminar of Science and Applied Technology (ISSAT 2020), USA, 2020.
11. D. D. Li and X. Y. Liu, “Research on MVP design pattern modeling based on MDA,” Procedia Comput. Sci., vol. 166, pp. 51–56, 2020.
12. A. L. Garcia, “Scrum-Based Application for Agile Project Management,” J. Softw, vol. 15, pp. 106–113, 2020.
13. M. Raczkowski, SCRUM Methodology for Running IT Projects in Dispersed Teams with Little Experience, USA, 2021.