IJournals: International Journal of Software & Hardware Research in Engineering
ISSN-2347-4890
Volume 5 Issue 1 January, 2017

Selective Software Project Management Methodology

Mohammed Awad[1]; Hisham Khdair[2]; Eman Alajrami[3]; Mohammed Manahi[4]

Faculty of Information Technology, University of Palestine[1,3,4]; Faculty of Technology and Information Science, University Kebangsaan Malaysia[2]

m.awad@up.edu.ps[1]; h.s.khdair@hotmail.com [2]; e.alajrami@up.edu.ps[3]; m.menahi@gmail.com[4]

ABSTRACT

The selection of a proper management methodology plays a central role in releasing a successful software project. Selective Software Project Management Methodology (SSPMM) is a framework that enables software personnel to select the methodology that suits their software project. It fills the gap of controlling and progress tracking related to software projects according to the chosen development methodology. Furthermore, it allows a software project manager to schedule tasks in each software development phase. In addition, it monitors how a certain task is accomplished and whether it succeeds or fails, and observes the time required for each task according to the timetable determined previously. SSPMM is built based on a combination of two Agile methodologies: XP and Scrum. In addition to realizing the SSPMM by developing a prototype, the framework was tested and evaluated based on both technical and end-user perspectives.

Keywords: Selective methodology, software methodology, project management.

1. INTRODUCTION

Software systems are becoming deeply embedded in every aspect of human lives and have become pervasive in commerce, culture and everyday activities [1, 2]. Software systems have appeared early to solve simple problems, then, to solve more complex problems such as performing complex medical operations [3] and automating nuclear power generation. Therefore, there is an increasing demand to well-engineer software applications to meet the challenges of the twenty-first century [4].

In term of project management, small and large software houses around the world choose web-based applications to manage their projects. A web-based project management software brings more usability, convenience and profitability to the companies what so ever their fields of production are.

The software engineering term in general means building well-engineered software applications based on systematic approaches that encompass a broad set of tasks included in the phases: requirements gathering and analysis, modeling and design, implementation, testing, and support of software systems to ensure that they are reliable for usage and solve the problem they were built for.

Using web-based technologies, a project management software can provide a set of facilities and services that make web-based project management software tools collaborate easily and updated automatically to increase the productivity with less effort than traditional ways.

Many projects have management problems without using or adopting systematic approaches and methodologies. This causes lack of managerial controlling of project progress in different phases [5]. In addition, team members may be located in independent locations or they have issues in face to face meetings. Therefore, there is a gap in the field of project management that is based on systematic approaches and methodologies, this gap could be filled by defining a framework and realizing this framework by developing a prototype that is built considering systematic approaches and methodologies. This prototype allows a project manager to schedule tasks in each phase, monitor and observe how tasks are whether accomplished successfully or fail; it enables the manager to estimate time required to each task in each phase according to some adopted systematic approaches and methodologies. In addition, it allows team members of a project to communicate easily to solve problems that may face them during the project life cycle [6].

2. SSPMM FOR MANAGING SOFTWARE PROJECTS

The SSPMM framework concepts are dedicated for the management of methodology-based software projects and people involved with software development using universally defined development methodologies such as Srcum and Extreme Programming (XP) methodologies, which are based on general agile development methodology.

The significance of an SSPMM system is providing a built-in and ready-to-use systematic approaches and methodologies for managing software projects online via a web-based application. It allows project managers to choose one of agile-based or one of System Development Life Cycle (SDLC-based) methodologies (if added) to improve productivity, save time and effort, and to be efficient as the first choice that replaces traditional ways.

Using SSPMM, project managers are being kept in the right track and the software project is being systematically progressed to reach the delivery phase with decreasing the failure possibilities during the project life cycle.

SSPMM is based initially on a combination of two Agile methodologies: XP and Scrum, XP methodology usually tends to fit technical aspects of software development [7-9], while Scrum tends to fit managerial aspects of software development [8-12]. Combining the two methodologies achieves developing the SSPMM prototype considering two concerns (clear-vision about technical development progresses and having a well-defined managerial control over technical development) to build a robust software project.

3. SSPMM'S PROTOTYPE

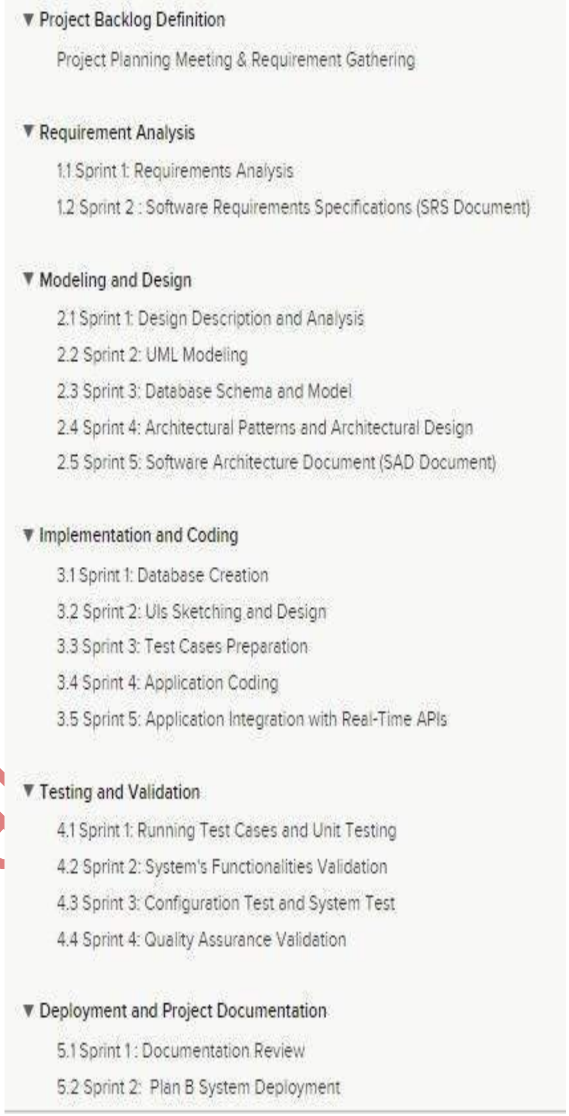
For the purpose of realizing SSPMM, a web based prototype system is developed to proof the SSPMM concept. The development of SSPMM prototype has considered some steps that should be followed by industry experts who want to realize SSPMM in the future. The following phases are followed in realizing SSPMM:

a) **Requirements Gathering and Specifications:** SSPMM requirements are gathered from people involved in software development, these requirements are about what is expected by SSPMM system to serve their aims including people who adopt the SDLC-based methodologies and the others who adopt the Agile-based methodologies, in addition to other methodologies if needed to be considered. After that, the scrum master and the team start prioritizing requirements according to its importance and its significant impact on the SSPMM prototype, and then they create a prioritized requirements list called project backlog (project backlog is initial due to the possibility of change on requirements). Furthermore, the team starts splitting the project backlog into sprints – typically each sprint takes two to four weeks, and then allocate tasks to the sprints. The scrum master and the team estimate the required time for the whole project and for each sprint including its tasks. Each sprint starts with a sprint planning meeting, which identifies time and tasks resources for that sprint. For the first set of sprints for the SSPMM, team start negotiation about the priorities of requirements, what is in scope and what is out. After negotiation, the team have to deliver a Software Requirements Specifications (SRS) document.

b) **Analysis and Design:** the analysis and design is the second set of sprints in the project backlog, the SSPMM team starts analyzing and designing the project backlog considering Unified Modeling Language (UML) diagrams including and not limited to some or all of:

usecase diagrams, class diagram, object diagrams, activity diagrams, sequence diagrams, collaboration diagrams, statechart diagrams, component diagrams and deployment diagrams. In addition to UML diagrams, the team designs the database schema and database modelling, the SSPMM team have to ensure that these diagrams illustrate the actual requirements SSPMM system is supposed to do. The SSPMM team has to attend daily meeting called daily standup meeting, and discuss what is done, what is not and what are the difficulties that faced each member in the team. Based on that, the scrum master assures the correctness of analysis and its diagrams, the design specialist in SSPMM team starts sketching the User Interfaces (UIs) and the team identifies the system workflow processes, then all sketches and workflows are delivered to scrum master [10, 11]. The scrum master writes a report which encompasses the diagrams and sketches that are delivered to the product owners (in SSPMM prototype, the product owners are people involved in software development) to receive constructive feedbacks from them. The SSPMM team repeats meetings and deliver reports until all prototype tasks are well-defined. After that, the SSPMM team and the product owners reach common understandings of all prototype specifications, and the preliminary design is created and the implementation phase starts.

c) **Coding and Implementation:** the code is written by converting analysis and design models to actual lines-of-code, whereas, before starting implementation and coding, the test cases have to be prepared. The sprint planning meeting identifies the test cases before starting coding, the scrum master recognizes test cases and directs the team to establish them [7, 13]. Once the test cases are prepared and ready-to-use, the team starts coding and implementing the system. After finishing the implementation, the whole code has to be reviewed before starting testing to ensure that there are no syntax errors and it is clean and ready for testing.



▼ Project Backlog Definition
- Project Planning Meeting & Requirement Gathering

▼ Requirement Analysis
- 1.1 Sprint 1: Requirements Analysis
- 1.2 Sprint 2 : Software Requirements Specifications (SRS Document)

▼ Modeling and Design
- 2.1 Sprint 1: Design Description and Analysis
- 2.2 Sprint 2: UML Modeling
- 2.3 Sprint 3: Database Schema and Model
- 2.4 Sprint 4: Architectural Patterns and Architectural Design
- 2.5 Sprint 5: Software Architecture Document (SAD Document)

▼ Implementation and Coding
- 3.1 Sprint 1: Database Creation
- 3.2 Sprint 2: UIs Sketching and Design
- 3.3 Sprint 3: Test Cases Preparation
- 3.4 Sprint 4: Application Coding
- 3.5 Sprint 5: Application Integration with Real-Time APIs

▼ Testing and Validation
- 4.1 Sprint 1: Running Test Cases and Unit Testing
- 4.2 Sprint 2: System's Functionalities Validation
- 4.3 Sprint 3: Configuration Test and System Test
- 4.4 Sprint 4: Quality Assurance Validation

▼ Deployment and Project Documentation
- 5.1 Sprint 1 : Documentation Review
- 5.2 Sprint 2: Plan B System Deployment

Fig 1: Combining XP and Scrum methodologies in SSPMM prototype

d) **Testing and Validation:** the final set of sprints in the project backlog is testing the code and the implementation of tests of the SSPMM prototype via test cases were prepared before implementation. Test-first development is one of the most important innovations in XP. Instead of writing some code and then writing tests for that code, you write the tests before you write the code. This means that developers can run the tests as the code is being written and discover problems during development. Testing the SSPMM prototype uses pair programming principle, that means each two developers in SSPMM form a set, each one in the set tests the other developer’s code and vice versa. During the daily standup meeting, the SSPMM team discuss testing progress and how to fix bugs that are showed up while testing. Validation process determines if the SSPMM prototype: complies with the requirements that are specified by the product owner, performs functions for which it is intended, and meets the product owner's expectations.

Figure 1 describes the combining process of XP phases into Scrum sprints and each phase of the two methodologies combined with each other.

4. DESIGN OF SSPMM PROTOTYPE

SSPMM was modeled using Unified Modelling Language (UML) version 2.1, designed using responsive design technologies, and implemented using PHP 5.6 - Laravel framework version 4.2. The following table and figures summarize the design activities done to develop the prototype:

- Table 1 shows actors of SSPMM prototype and a description of each role and its functionalities
- Figure 2 shows usecases' diagram for the group "user" of SSPMM prototype
- Figure 3 shows usecases' diagram for the group "team member" of SSPMM prototype
- Figure 4 shows usecases' diagram for the "administrator" of SSPMM prototype
- Figure 5 shows usecases' diagram for group "manager" of SSPMM prototype
- Figure 6 describes the complete sequence activities for performing required tasks assigned to a team member
- Figure 7 shows the database model for SSPMM prototype

Table 1. System actors and roles description

Actor	Role Description	Functionalities
Administrator	An actor who is responsible for system administration.	▪ Manage website contents by adding, editing and removing contents. ▪ Manage groups by adding, editing and removing groups, managers and team members
Group User	A generalized actor who represents the manager's and team member's functionalities in common.	▪ Send and receive tasks attachments within a group. ▪ Send and receive messages and notifications within a group. ▪ Post and comment in the group's wall. ▪ Chat with group manager and member within a group. ▪ Edit settings and update information.
Group Manager	An actor who is responsible to manage the group of the team members by assigning tasks and controlling the work progress.	▪ Invite project team members to group and allocate resources. ▪ Manage group content by adding, editing and removing contents and change/update group settings and information. ▪ Scheduling project phases, tasks and project timetable. ▪ Assign tasks and its deadlines and the team member who is assigned to perform it. ▪ Control and observe each task and each team member deliverables
Group Team Member	An actor who is responsible for carrying out tasks and perform required work.	▪ Perform work and carrying out tasks that are assigned by the manager. ▪ Respond to manager messages and submit required work. ▪ Send submissions of accomplished work and tasks.

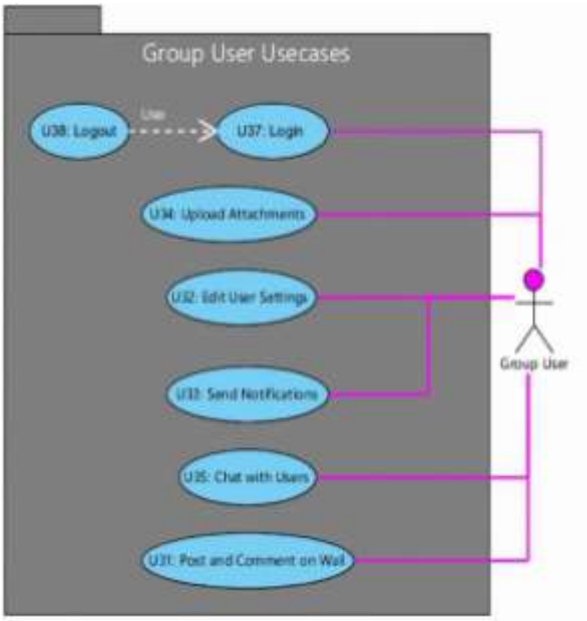


Fig 2: Group "User" usecase diagram

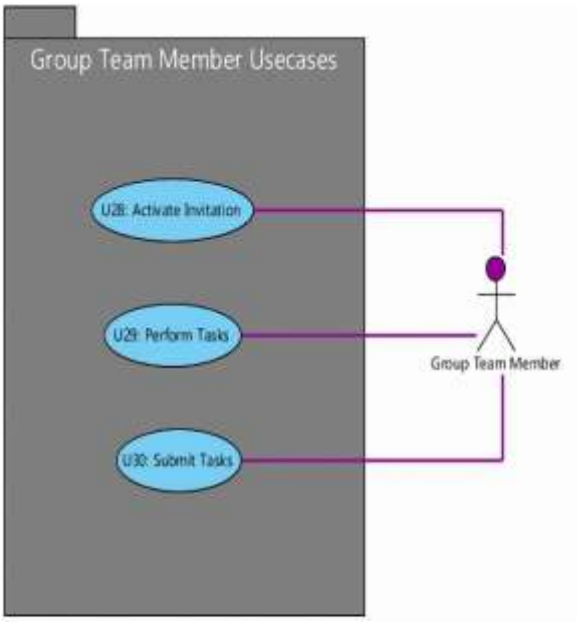


Fig 3: Group "Team Member" usecase diagram

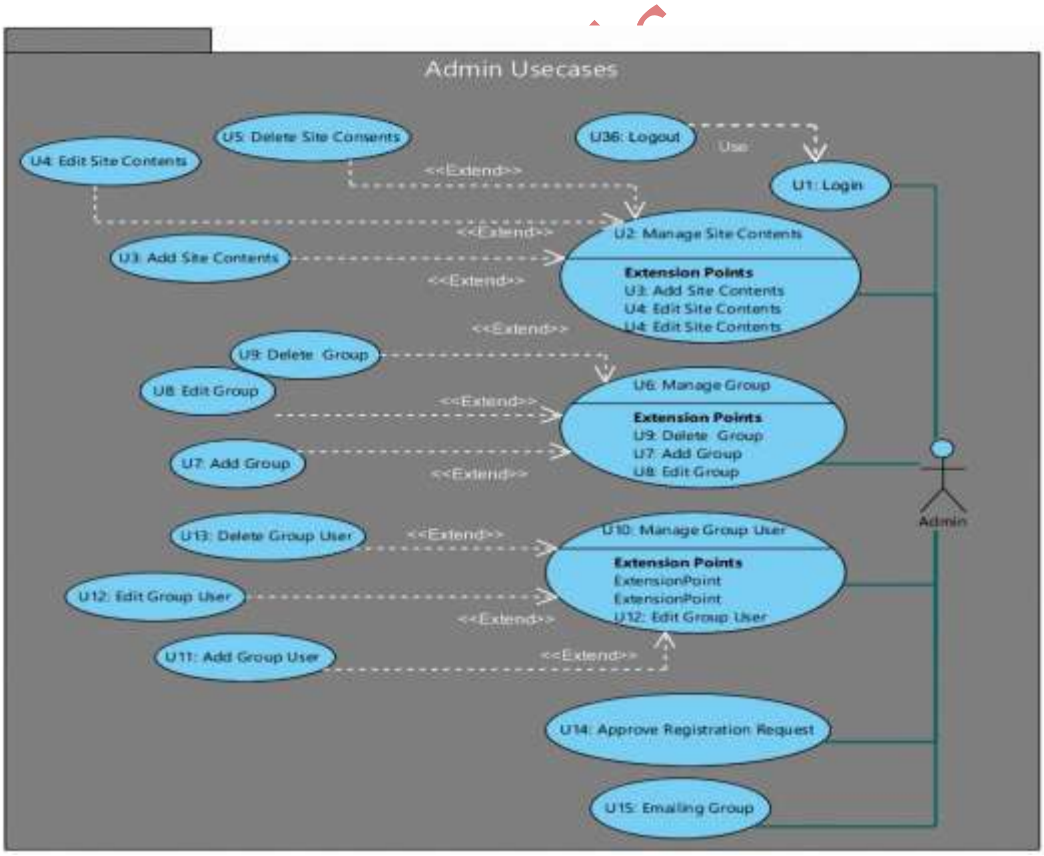


Fig 4: Admin usecase diagram

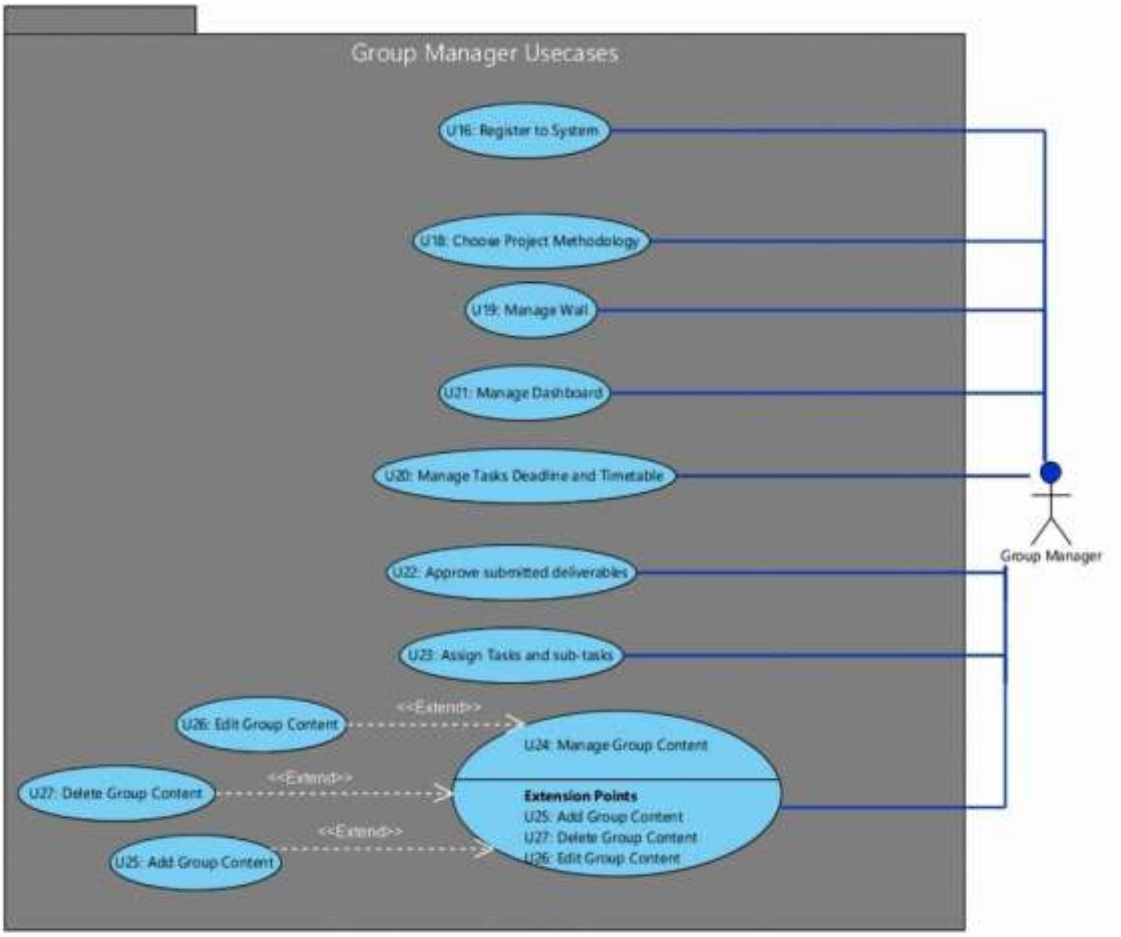
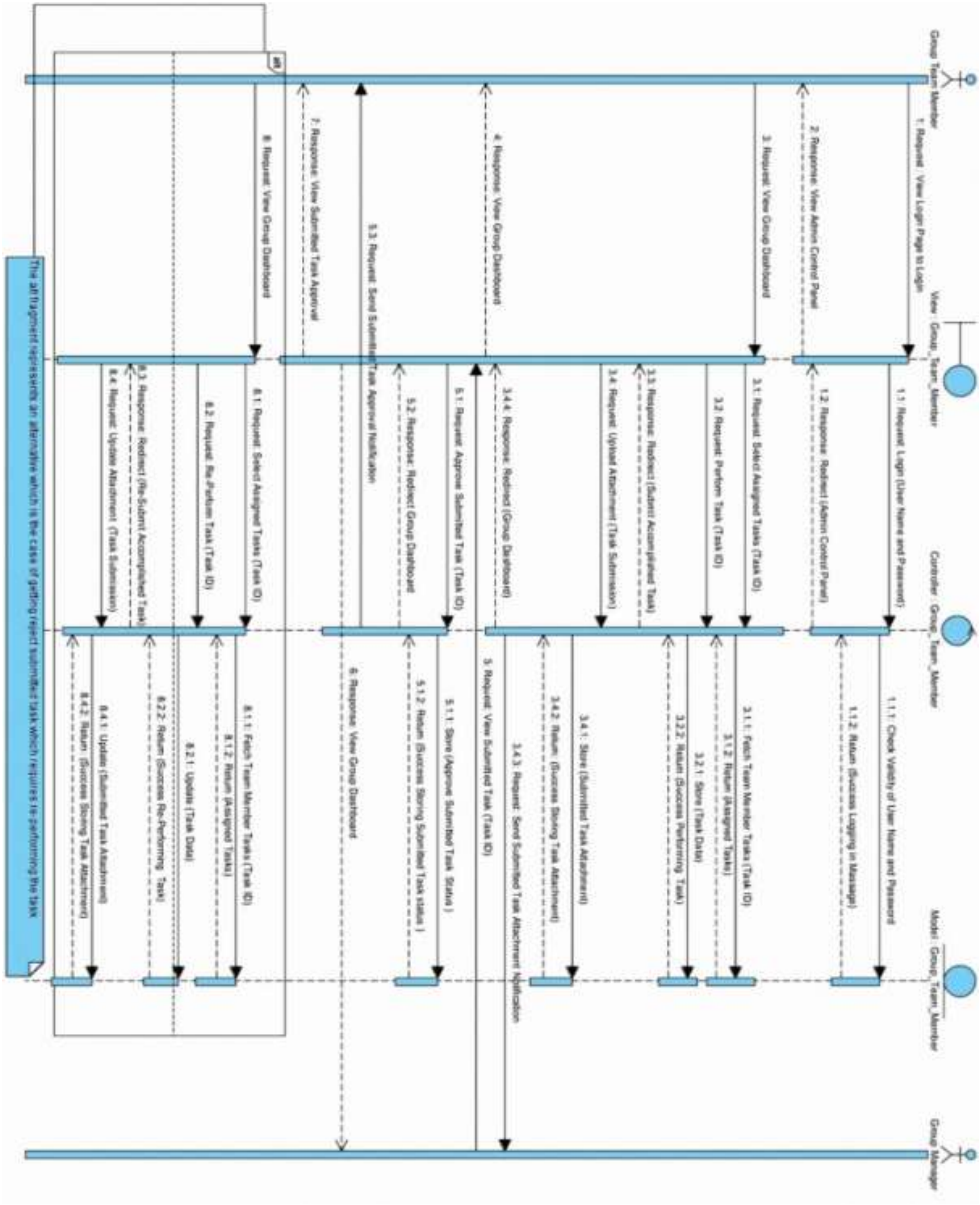


Fig 5: Group "Manager" usecase diagram


Group Team Member
View : Group_Team_Member
Controller : Group_Team_Member
Model : Group_Team_Member
Group Manager
1: Request : View Login Page to Login
1.1: Request: Login (User Name and Password)
1.1.1: Check Validity of User Name and Password
1.1.2: Return (Success Logging in Massage)
1.2: Response: Redirect (Admin Control Panel)
2: Response: View Admin Control Panel
3: Request: View Group Dashboard
3.1: Request: Select Assigned Tasks (Task ID)
3.1.1: Fetch Team Member Tasks (Task ID)
3.1.2: Return (Assigned Tasks)
3.2: Request: Perform Task (Task ID)
3.2.1: Store (Task Data)
3.2.2: Return (Success Performing Task)
3.3: Response: Redirect (Submit Accomplished Task)
3.4: Request: Upload Attachment (Task Submission)
3.4.1: Store (Submitted Task Attachment)
3.4.2: Return: (Success Storing Task Attachment)
3.4.3: Request: Send Submitted Task Attachment Notification
3.4.4: Response: Redirect (Group Dashboard)
4: Response: View Group Dashboard
5: Request: View Submitted Task (Task ID)
5.1: Request: Approve Submitted Task (Task ID)
5.1.1: Store (Approve Submitted Task Status)
5.1.2: Return (Success Storing Submitted Task status)
5.2: Response: Redirect Group Dashboard
5.3: Request: Send Submitted Task Approval Notification
6: Response: View Group Dashboard
7: Response: View Submitted Task Approval
alt
8: Request: View Group Dashboard
8.1: Request: Select Assigned Tasks (Task ID)
8.1.1: Fetch Team Member Tasks (Task ID)
8.1.2: Return (Assigned Tasks)
8.2: Request: Re-Perform Task (Task ID)
8.2.1: Update (Task Data)
8.2.2: Return (Success Re-Performing Task)
8.3: Response: Redirect (Re-Submit Accomplished Task)
8.4: Request: Update Attachment (Task Submission)
8.4.1: Update (Submitted Task Attachment)
8.4.2: Return: (Success Storing Task Attachment)
The alt fragment represents an alternative which is the case of getting reject submitted task which requires re-performing the task

g 6: Sequence diagram for performing tasks assigned to a team member

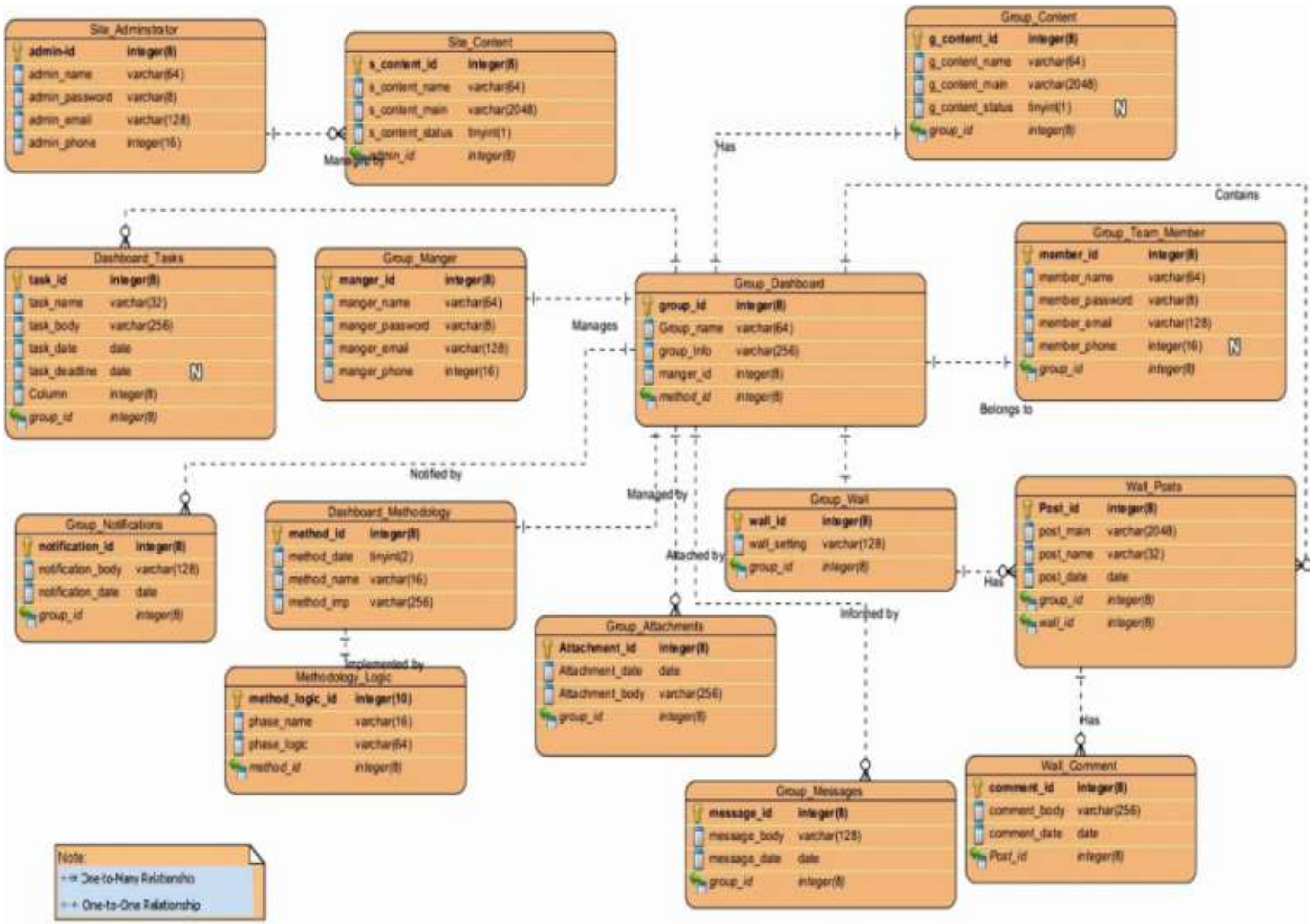


Fig 7: Database model diagram

5. THE ARCHITECTURE OF SSPMM PROTOTYPE

The architecture of SSPMM prototype is built based on Resource Oriented Architecture (ROA), which is a specific set of guidelines of an implementation of the REST-style architecture [14]. Based upon the concept of resource, each resource is a directly accessible distributed component that is handled through a standard common interface making possible resources handling. Representational State Transfer (RESTFul) platforms based on REST development technology enable the creation of ROA.

The REST architecture style is designed using the following components:

- Clients: Clients initiate a request.
- Servers: Servers process and respond to requests.
- Resource: Resource is a concept to describe things (i.e. if the application is a project management web application, the resource will be the tasks).

Figure 8 shows the REST architecture style description [15, 16] and how the request processed when REST is implemented in web applications.

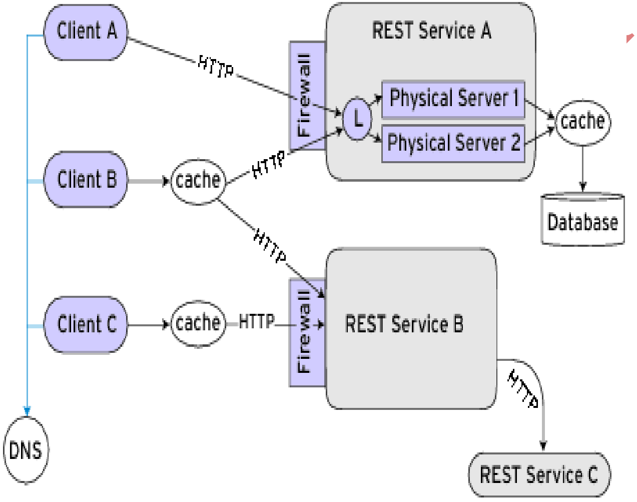


Fig 8: REST architectural style

In addition, MVC - Model-View-Controller – is adopted. It is a design pattern for the architecture of web applications. It is a widely adopted pattern, across many languages and implementation frameworks, whose purpose is to achieve a clean separation between each component responsibilities (presentation (UI), application logic, and data model).

The MVC design pattern encompasses three components [17]:

- Model: The model uses persistent storage mechanism (such as a database) to store data, manipulate, and retrieve the data. It encapsulates the appropriate data, and exports procedures that perform application-specific processing. The model also provides functions to access its data that are used by view components to acquire the data to be displayed.
- View: The view present information to the user. Different views present the information of the model in different ways. Each view defines an update procedure that is activated by the change propagation mechanism. When the update procedure is called, a view retrieves the current data values to be displayed from the model, and puts them on the screen.
- Controller: The controller processes and responds to events, typically user actions, and may invoke changes on the model and view. Also the Controller can send commands to the model to update the model's state. It can also send commands to its associated view to change the view's presentation of the model.

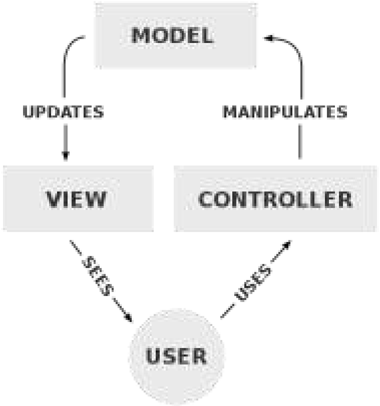


Fig 9: MVC Design Pattern

6. TESTING OF SSPMM PROTOTYPE

Table 2 shows testing process in SSPMM prototype. It addresses two project phases of testing described as following:

Table 2. Testing plan

Project Phase	Test Level/Type	Affected Area
Development Phase	Unit Testing/White Box Testing	Methods and code-lines of the SSPMM system
Integration Phase	System Testing Integration Testing/ White and Black Box Testing	Integration of third-party application with SSPMM system

Agile testing is adopted, which is a testing practice

for a project using agile software development methodologies, incorporating techniques and methods such as extreme programming (XP). It considers emphasizing the test-first design paradigm.

The testing criteria of SSPMM prototype included two levels of testing which are unit-testing and integration testing.

Table 3 shows the results of measuring passed and failed tests done to validate the SSPMM prototype:

Table 3. Testing checklist

No.	Parameter	Description	Passed	Failed	N/A
1	Browser Compatibility	Application performance on a variety of browser types and configurations.	✓		
2	Functional Correctness	Application functions correctly including validating links, calculations, displays of information, and navigation.	✓		
3	Integration	The integration between browsers and servers, applications and data, and hardware and software.	✓		
4	Usability	The overall usability of a web page or a web application, including appearance, clarity, and navigation	✓		
5	Security	The adequacy and correctness of security controls, including access control and authorizations.	✓		
6	Performance	The performance of the web application under load.			✓
7	Verification of code	The Validation of the code used in building the web application) has been used in a correct manner.	✓		
8	Easy to use	The application is easy to use and User Interfaces (UI) are user friendly	✓		
9	E-mail functions	Email services Functioning as well as it should	✓		
10	File downloads	Ability of uploading and downloading attachments and files	✓		
11	Authorization levels	Ability of the application to restrict certain transactions only to those users who have a certain level of authorization	✓		

7. EVALUATION

A group of 10 technical persons is selected from different IT companies to evaluate the research prototype. Two project managers, two analysts and designers, four developers, and two testers. The technical evaluation of SSPMM prototype evaluates the percentage of satisfaction level of the respondents. Table 4 shows the average level of satisfaction among the 10 respondents.

Table 4. Technical Evaluation

No.	Parameter	Description	0-25%	25-50%	50-75%	75-100%
1	Browser Compatibility	Application performance on a variety of browser types and configurations.				✓
2	Functional Correctness	Application functions correctly including validating links, calculations, displays of information, and navigation.				✓
3	Integration	The integration between browsers and servers, applications and data, and hardware and software.			✓	
4	Usability	The overall usability of a web page or a web application, including appearance, clarity, and navigation			✓	
5	Security	The adequacy and correctness of security controls, including access control and authorizations.			✓	
6	Performance	The performance of the web application under load.			✓	
7	Verification of code	The Validation of the code used in building the web application) has been used in a correct manner.				✓
8	Easy to use	The application is easy to use and User Interfaces (UI) are user friendly			✓	
9	E-mail functions	Email services Functioning as well as it should			✓	
10	File downloads	Ability of uploading and downloading attachments and files			✓	
11	Authorization levels	Ability of the application to restrict certain transactions only to those users who have a certain level of authorization				✓

The overall evaluation process showed that the level of satisfaction of SSPMM prototype was accepted by the respondents with an average percentage of 71%.

8. CONCLUSION

Project management is critical for developing successful software projects. Enabling the flexible selection of a proper management methodology and its phases to suit the project to be developed; increases the success possibilities of the project. SSPMM is proved in this paper to be a good choice for managing software projects. It is a web application which was designed, implemented, tested and evaluated successfully. It was accepted by evaluation respondents since it provides an effective way for managing software projects.

9. REFERENCES

[1]. A. Badoiu, S. Petrescu, V. Vlad, and A. Botu, "Information System for the Management of the Health Services in Romania," Proceedings of IEEE International Conference for Robotics Automation, Quality and Testing., 2008.

[2]. A. R. Hevner and S. T. March, "The Information Systems Research Cycle," MIS Quarterly (November), vol. 36, pp. 111 - 113., 2003.

[3]. J. L. NITZKIN and C. BUTTERY, "Public Health Information Infrastructure," IEEE Engineering in Medecine, 2008.

[4]. J. Henrard, J. M. Hick, P. Thiran, and J. L. Hainaut, "Strategies for data reengineering," Proc. WCRE02, IEEE Computer Society Press, 2002.

[5]. R. Goel, M. C. Govil, and G. Singh, "A secure software design methodology," in 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2016, pp. 2484-2488.

[6]. S. W. A. Rizvi, V. K. Singh, and R. A. Khan, "Revisiting software reliability engineering with fuzzy techniques," in 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 2016, pp. 1037-1042.

[7]. A. Marchenko and P. Abrahamsson, "Scrum in a Multiproject Environment:An Ethnographically-Inspired Case Study on the Adoption Challenges," Proceedings of Agile 2008 Conference, 2008.

[8]. E. Uy and N. Ioannou, "Growing and Sustaining an Offshore Scrum Engagement," Proceedings of Agile 2008 Conference, 2008.

[9]. K. Schwaber. (2009, 2/3/2009). Scrum Development Process. Available: http://jeffsutherland.com/oopsla/schwapub.pdf.

[10]. B. Barton and E. Campbell, "Implementing a Professional Services Organization Using Type C Scrum," Proceedings of 40th Hawaii International Conference on System Sciences, 2007.

[11]. M. Cristal, D. Wildt, and R. Prikladnicki, "Usage of SCRUM Practices within a Global Company," Proceedings of IEEE International Conference on Global Software Engineering, 2008.

[12]. S. H. Rayhan and N. Haque, "Incremental Adoption of Scrum for Successful Delivery of an IT Project in a Remote Setup," Proceedings of Agile 2008 Conference, 2008.

[13]. K. H. Judy and I. Krumins-Beens, "Great Scrums Need Great Product Owners: Unbounded Collaboration and Collective Product Ownership," Proceedings of 41st Hawaii International Conference on System Sciences, 2008.

[14]. K. Dar, A. Taherkordi, H. Baraki, F. Eliassen, and K. Geihs, "A resource oriented integration architecture for the Internet of Things: A business process perspective," Pervasive and Mobile Computing, vol. 20, pp. 145-159, 7// 2015.

[15]. R. P. V. Chander, "Web Services for the Internet of Things -- A Feasibility Study," in 2016 International Conference on Distributed Computing in Sensor Systems (DCOSS), 2016, pp. 125-126.

[16]. S. K. Datta, C. Bonnet, R. P. F. D. Costa, and J. Härri, "DataTweet: An architecture enabling data-centric IoT services," in 2016 IEEE Region 10 Symposium (TENSYMP), 2016, pp. 343-348.

[17]. P. P. Churi, S. Wagh, D. Kalelkar, and M. Kalelkar, "Model-view-controller pattern in BI dashboards: Designing best practices," in 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 2016, pp. 2082-2086.