+MRT: Centralized Web Based Application for Managing Repetitive Tasks

Naveed Ullah[a]*, Rizwan Mehmood[b]

[a] Lahore Leads University, Lahore, Pakistan 8F, M block Gulberg 3, Lahore and 54000, Pakistan
[b]Lahore Leads University, Askari 11 Sector B Building 9 Flat 2c, Lahore and 54000, Pakistan
[a]Email: naveedpaen@gmail.com, [b]Email: dr.mahmood@leads.edu.pk

Abstract

Abstract—Task management is broadly classified as a process of managing and assigning different tasks to co-workers, employees or subordinates throughout the life cycle of a project via project managers. Tasks can be of various natures but more specifically we are referring here tasks that are used in the software industry. In this modern world, why a project manager has to assign a repetitive task to its team members repeatedly when a new project arrives. This research paper particularly addresses this issue of automatic task management. In this research we have investigated the problems regarding task management, that can occur during the life cycle of a project and presented a web application MRT which can be effectively served as a solution to those problems. Existing applications like Jira and Trello etc. are helping project managers to automate this process. However, we argue that they are still lacking in few aspects. After examining the existing tools, we felt that there are some problems in automatic task management which are not solved yet. We have solved the problems of task management via MRT. We claim that project managers can use MRT for automatic tasks assignments. MRT novel features, therefore, eliminates the need for human effort for repetitive tasks management. Another aspect of Task management is to provide global access to information regarding tasks status to different levels of employees in hierarchy such as manager, CEO, that helps them regarding taking strategic decisions about employees and further projects. MRT can be used as a safeguard against delays and missed deadlines. More precisely our contributions are i) An algorithm for assigning automatic tasks to employees based on their skill set. ii) Tracking of task dependencies that leads to successful project completion iii) Sub Tasks dependencies tracking that are essential to consider, for declaring a task as completed, pending or not completed. iv) Notification management for timely review of tasks. v). Task Tracking at various levels. vi) Priority-based task assignment vii) Additional feature of document storage viii) Creation of Executive Dashboard of the project to visualize the important metrics. ix) Usage of state of the art ASP.Net Web APIs which ensure the use of mobile apps for MRT x) Fresh API Tokens are sent on each call to enhance the security of MRT. xi) The creation of web services that allows integration of MRT data into various devices such as mobiles, tablets etc.

----

* Corresponding author.

1. Introduction

Despite the fact that there is an enormous increase in project management methods and task-management technologies, still projects do not complete in time and there is mismanagement in tasks handling. Another important factor is enormous growth in a number of projects that a company or software house is handling at a point in time. Over the years there is significant growth in software development projects and companies are emerging on the scene and problems related task management also increases correspondingly. Every company in this world works on projects. Besides working, it always seeks new projects all the times for its sustainability. The growth of a company depends on the successful and timely completion of projects. A project is a collection of multiple tasks which are assigned to different employees in a company. Typically, every new project resembles the previous one. Companies have many employees. Most employees work on projects which are same in nature, in other words, the tasks of each project are repeated again and again for many employees for various projects. Just to elaborate this a lecturer in a university teaches the same course to a group of students in new upcoming sessions. To further elaborate our argument, for instance, consider a software house, where static websites are developed. A static web application comprises of four tasks, which means, four employees are required to complete four different tasks of a project. So, project manager needs four employees with four different skills to create a static website. The first employee would purchase a domain, the second one would design the Website, the third one would write HTML, CSS and the fourth one would test the website once it is completed and hosted. So, every employee has skills in a specific area. Once this project gets completed then the company would have many more projects of the same nature. Therefore, this research is based on solving this dilemma as to why a projects manager needs to assign the same task again and again to an employee although we already know the expertise of each employee and we already know most of the projects are of same types which always need four employees to be done. Generally, companies use task management applications like Jira and Trello where tasks cannot be assigned automatically. Therefore, a system is required which automatically assigns tasks to its relevant employee, and human assigning should be avoided for repetitive tasks. While in case we have a new task, then that specific task can be manually assigned by the project manager to its employees. The rest of paper is organized as follows. Section II explains a literature review. Section III explains Architectural Overview. Section IV explains Methodology of MRT with four different scenarios. Section V explains some code. Section VI is about MRT features. System Evaluation is elaborated in Section VII followed by Future Work in Section VIII. We end with a short conclusion in Section X.

2. Literature Review

This section highlights some of the previous and existing practices to solve the problem of Task management. We will elaborate on how existing tools are working and how MRT is different from existing tools and approaches. Previously Task management systems were dependent on historic data of the company and on the basis of that old data, decisions were being taken. OLAP and Data Mining techniques are used to decide which employee should be assigned a task. Historical data is evidently the prime source of data for predicting future and it is usually structured in database in the form of rows and columns for easy access. This makes it ideal for

all kinds of numerical analyses, including online analytical processing (OLAP) and data mining. Actually, it would check the history of a task, and if Task A is 80% assigned to Employee A and 20% assigned to Employee B, then next time Task A would be assigned to Employee A. Using this approach, the accuracy would never be 100%. Secondly, if a company is recently started, which have no historical data, in that case, there is no possibility to assign task automatically on the basis of historic data [1,2,9,10,11,12]. However, MRT never depends on historic data, you have to just once explain the expertise of the employees to the system after that 100% accuracy rate is granted. Even the most used project management systems like Jira and Trello do not facilitate auto tasks. Recently there are two famous applications used for project management worldwide. Firstly, Jira [3] is a proprietary issue tracking product developed by Atlassian that allows bug tracking and agile project management. According to Atlassian, Jira is used for issue tracking and project management by over 75,000 customers in 122 countries. Some of the organizations have used Jira at some point in time for bug-tracking and project management. Secondly, Trello [4] is a web-based list-making application originally made by Fog Creek Software in 2011. In May 2016, Trello claimed that it had more than 1.1 million daily active users and 14 million total signups. As far as MRT is concerned, it provides the facility to manage auto tasks including all the other facilities which Jira and Trello provide.

3. Architectural Overview

Before delving into the design of the individual components, Figure 1 provides an architectural overview of all the components that constitute the entire system. Our server-side is composed of three-layer architecture. Data Access Layer, Repository Layer, and Business Logic Layer. Repository Layer contains methods that help the business layer to connect with the database and perform CRUD operations. Web API Controller receives HTTP requests and returns data in the form of JSON which can be used by Web and mobile client. Currently, we have the only web-based application of MRT.
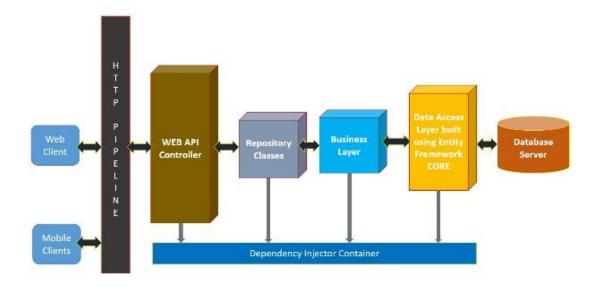

Figure 1: Architectural Overview

4. Methodology

4.1. How Tasks will be assigned Automatically?

Firstly, the expertise of each employee is declared in MRT Auto Tasks Module. E.g. a Company which creates static websites having four employees. Employee A can purchase a website, Employee B is able to design a website while Employee C has expertise in Coding HTML, CSS and JavaScript. Employee D can test the website as shown in Figure 2. Now a client Z needs a website, so four employees with four different expertise are required to complete this job which is already residing in the Auto Task List.
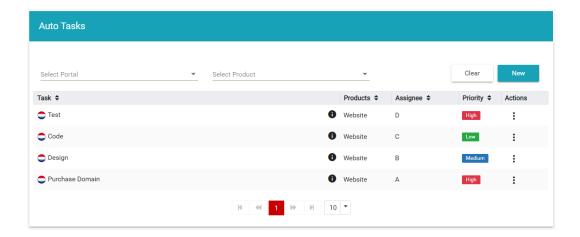

Figure 2: Auto Tasks List

As we create a new project for current client Z in MRT, it assigns tasks automatically to its relevant employee without assigning by the project manager as shown in Figure 3. Next time if the company receives a website creating project again from someone, project manager just has to create a project and tasks will automatically be assigned to its relevant developers. This is what we are achieving in MRT.
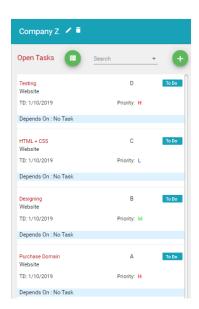

Figure 3: Tasks Creation from Auto Tasks

4.2. How Sub Tasks will be Assigned Automatically?

Auto tasks and Manual tasks can have subtasks. Basically, subtasks are part of Auto Tasks and Manual tasks. let's understand it by example. If we have created a new project by name of Project Z for our new client. and that project contain 4 tasks (A, B, C, D) and one subtask, As shown in Figure 4, While MRT already contains Task A, B, C along with subtask 1 in the Auto Tasks list, so why would we create these tasks for our new client again. Hence there is no need to create these tasks again, therefore three tasks (A, B, C) with one subtask will be taken from Auto Task Template and forth one Task D will be inserted manually by the operator. The Figure 4 shows how a sub-task along with auto task help us from repetition.
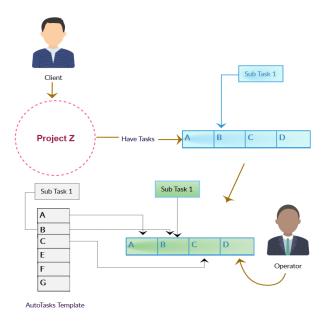

Figure 4: Automatic Sub Tasks generation in MRT

4.3. Priority Management

Every task has priority which means some tasks can have more importance than other tasks, and those tasks should be completed quickly compared to other tasks. In MRT there are three types of Priorities, High, Medium and low. A task with High Priority needs to be done very quickly. Medium priority tasks are not that much urgent, while less priority tasks can be done according to the ease of a developer. In MRT there are grids which show the lists of tasks; therefore, every grid shows tasks on the basis of priority, which means high priority tasks will be seen on the topmost, and then medium and low priority tasks respectively. In Figure 5, H with red color is high priority tasks, while M means Medium and L in blue color means Low priority tasks. If there are multiple tasks with high priority then MRT will show those tasks on top which have less time in due date.
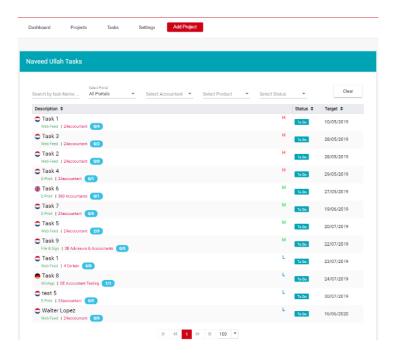
Figure 5: Tasks Grid

4.4. Scrum Board Management

Scrum Board is a summarized form of tasks that helps teams to make Sprint Backlog items visible. The board can take many virtual and physical forms but it performs the same function regardless of how it looks. The board is updated by the team members and shows all tasks that need to be completed for the current Sprint. When using Scrum, we can make the sprint columns visible by putting it on a Scrum task board. A team updates the task board continuously throughout the sprint; if someone thinks of a new task, he writes a new card and puts it on the board. Either during or before the daily scrum, estimates can be changed up or down, and cards can be moved around the board as shown in Figure 6.
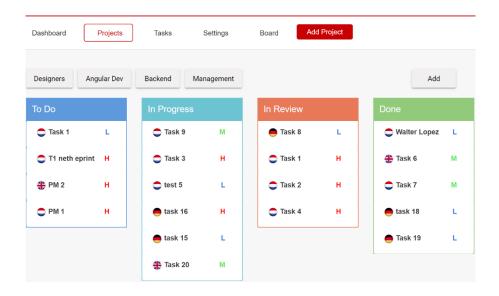

Figure 6: Scrum Board

Each row on the Scrum board is the user story, which is a basic unit of work, we encourage teams to use for their product backlog. During sprint planning, the team selects backlog items they can complete during the coming sprint. Each product backlog item is turned into multiple sprint backlog items and each of these is represented by one task card that is placed on the Scrum board. Each task starts on the Scrum task board in the "To Do" column.

5. Implementation

We are presenting here code in SQL Server stored procedure just for illustration of automatic task management in Figure 1. Tasks are created from Auto tasks. Basically, Auto-task behaves like a template, which can be used repeatedly. In line number 2, 3 all the auto tasks are selected from Autotask-table, while in line Number 4,5 those tasks are inserted against each project in a separate table called Task-table. Tasks residing in Task-table are actual targets which would be done by employees, from line number 6 to 11 Tasks dependency is managed, which means that a task can be dependent on another task.

```
1 select @MaxId=max(id)from superadmin.issues;
2 ;WITH NewIDs AS (SELECT OriginalID=T.Id, ReplacementID=ROW_NUMBER()
3 OVER (ORDER BY T.ID ASC) FROM ImProject.autotasks AS T)
4 INSERT INTO superadmin.issues(Id, ParentId, Name,ProductId,
5 AssignedToId,ProjectId,Assigner_Id)
6 SELECT N1.ReplacementID+coalesce(@MaxId, 0),
7 case when T.ParentId=0 then 0 else ISNULL(N2.ReplacementID, 0)
8 +coalesce(@MaxId,0)
9 Name=T.Name,ProductId=T.ProductId, AssignedToId=T.AssignedToId,
10 ProjectId=@ProjectId,  FROM ImProject.autotasks AS T
11 INNER JOIN NewIDs AS N1 ON T.Id=N1.OriginalID
12 LEFT JOIN NewIDs AS N2 ON T.ParentId=N2.OriginalID
13 where PortalId=@Portal AND ProductId=@SelectedProductId
14 SET IDENTITY_INSERT superadmin.issues OFF
```

Figure 11: Store Procedure to Generate Task Automatically

6. MRT Features

We have developed this system using open source technologies. Our company, where author 1 is working is using this system. MRT is a web-based application allowing the company to know the overall progress of tasks assigned to the development team. For every client, the application creates a new project, which contains our products selected by the client. Each product has some specific tasks against it. Our application has the flexibility of assigning tasks automatically and manually to the relevant developer. Our system has a list of tasks as a template which are same for a specific product. As client selects our product, those template tasks are automatically get created for their respective developer. In addition, a project manager can attach any important file with the task, send emails to employees for their pending work daily, check Activity of major operations, can search a task easily. Further, the Executive Dashboard of the project visualizes the most important metrics
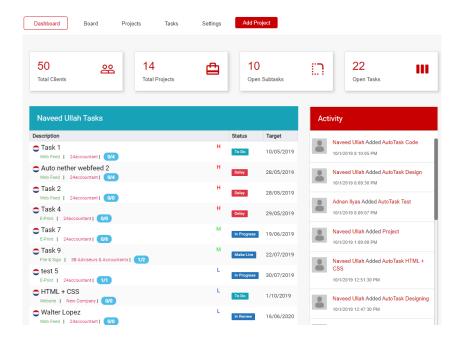
only which is shown in the Figure 7.

Figure 7: Dashboard

Security has been the main concern to our applications and it has become an absolute necessity. Our Web APIs are secured with a token system, which make sure high-level security. MRT is using token-based authentication consuming asp.net web API 2. In the token system, a request is sent from the client to an authentication server for verification of client credentials. If a user exists in the database, A token is provided by the server and that token is taken to web API which authorizes the correct token and gets access to methods resides in the controller as shown in the figure 8. In addition to security, web API also have the flexibly to be consumed by Mobile application. In the future, our focus would be to develop a mobile interface of MRT. We are achieving high performance and best architectural design by using the best tools to develop the system, namely C# [6], Angular [7] and SQL [8] technologies. Especially, the latest Angular framework for front end and ASP.net Web.
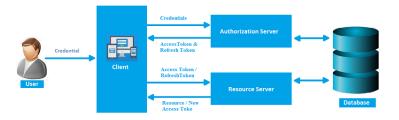

Figure 8: Token Based Authentication

API's boosted the speed of the system. C# is a multi-paradigm programming language developed by Microsoft within its.NET initiative. It is intended to be a simple, modern, general-purpose, object-oriented programming language. Angular is a structural framework for dynamic web apps. Structured Query Language (SQL) is a standard computer language for relational database management and data manipulation. SQL is used to query,

insert, update and modify data.

7. Evaluation

Jira is the best-selling software development application from Atlassian used by a lot of tech teams. it offers project planning and tracking features. Designed for agile software development teams, Jira offers various dashboards and overviews for scrum and agile techniques with 1,000 add-ons and extensions, Jira enables you to integrate its software with other 3rd party tools you might already use. Trello on the other hand – is largely designed for more general usages. Trello is a lightweight project management tool, owned by Atlassian too. Designed as a Kanban board tool, you can visualize your task flow of anything from software development projects, to marketing campaigns, and to holiday trips. MRT on the other hand – provides Auto Tasks Management system, and customization in the application according to the needs of a company. In terms of simplicity MRT is quite simple enough for its users, and so far used by many employees across three countries (UK, Netherland, Germany) Irrespectively all of them are project management tools, Jira, Trello, and MRT have some key differences. Since Jira was designed for software teams, it mainly targets software builders, developers, or project managers working on soft-ware projects. In contrast, Trello has a much broader target audience, as it offers any kind of project tracking. As far as MRT is concerned, it is the combination of Jira and Trello, additionally it provides complex operations in a simple and reliable form. Some key differences between MRT, Jira, and Trello are shown in Figure 9.

Feature	MRT	Jira	Trello
Automatically Assign Task	✓		
Sub Tasks	✓	✓	✓
Task Dependency	✓	✓	✓
Comments & Emails	✓	✓	✓
Scrum Board	✓	✓	✓
Customization	✓		
File Attachment	✓	✓	✓
Responsive	✓	✓	✓
Task Priorities	✓	✓	✓
Extensions		✓	✓
Check Activity	✓	✓	✓

Figure 9: Comparison between MRT Jira and Trello

8. Future Work

Mobile apps are becoming increasingly popular among businesses and service providers. There are almost no industry where mobile applications haven't proved their marketing value or established themselves fundamental for effective communication with target customers, while MRT does not have a mobile application for android and IOS operating systems. In the future, our high priority would be to develop mobile and smart watch interface as shown in figure 10. Secondly. we have advertised MRT locally, but in near future, we are hopeful that MRT would be used by different businesses around the world. Lastly, we will provide Services (API Services) for our clients so that they can integrate MRT in their own application.
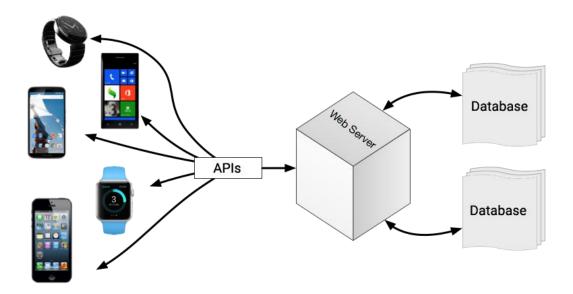
Figure 10: Future Work

Acknowledgements

9. Conclusion

Generally, a manager assigns tasks to its employees in an organization, and occasionally same task is needed in different projects to be assigned again and again to employees, which is time taking process for managers. To tackle this problem, we have decided to split a task into two types, first type is Manual task assignment by manager to its employees, and second type is Auto task which is generated and assigned automatically by the application to all employees. In this research paper we have integrated the ability of handling repetitive tasks automatically and this facility would help us to save time and human resources while assigning work in an organization.

References

[1]   Schunk, Lars K., Cong, Gao. (2009)."Automated Assignment of Em-ployees to Work Tasks."Master's Thesis, Aalborg University,2009.

[2]   Asbjorn Rolstadas, Iris Tommelein, Per Morten Schiefloe, Glenn Bal-lard, (2014) "Understanding project success through analysis of project management approach", International Journal of Managing Projects in Business, Vol. 7 Issue: 4, pp.638-660.

[3]   M.Cannon-Brookes.Jira. Internet: https://www.atlassian.com/software/jira [29 June 2019].

[4]   Joel Spolsky.Trello. Internet: https://trello.com/en [29 June 2019].

[5]   Dustin Moskovitz.Asana. Internet: https://asana.com [20 June 2019].

[6]   Microsoft   Contributors.   (2019,   April   14).   "What's   New   in   C#   7.0".   Available: https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-7 [20 June 2019].

[7]   Google   Contributors.   (2018,   April   14).   "Introduction   to   the   Angular   Docs".   Available: https://angular.io/docs [20 June 2019].

[8]   Microsoft   Contributors.   (2018,   April   14).   "Microsoft   SQL   Server".   Available https://searchsqlserver.techtarget.com/definition/SQL-Server [20 June 2019].

[9]   Bourne, L. and Walker, D. (2008), "Project relationship management and the Stakeholder Circle™", International Journal of Managing Projects in Business, Vol. 1 No. 1, pp. 125-130.

[10]   Vidal, L. and Marle, F. (2008), "Understanding project complexity: implications on project management", Kybernetes, Vol. 37 No. 8, pp. 1094-1110.

[11] Longman, A. and Mullins, J. (2004), "Project management: key tool for implementing strategy", Journal of Business Strategy, Vol. 25 No. 5, pp. 54-60.

[12] Kutsch, E. (2008), "The effect of intervening conditions on the man-agement of project risk", International Journal of Managing Projects in Business, Vol. 1 No. 4, pp. 602-610.