

TEAM PROJECT TRACKER

CMSE 473

Final Report of Term Project

Team members: Talal Mahdy - 147139

Adham Moshasha - 148387

Mohamed M. M. Balto - 147697

Abdoulgwad Elsheredi -147597

Supervisor: Assoc. Prof. Dr. Duygu Çelik Ertuğrul

Software Engineering Department

Eastern Mediterranean University

December 2017

ABSTRACT

This project aims to produce an Android Application that is able to manage all types of projects for teams. The application's main objectives are to allow the employees to join different teams, save the time of the employees, to make the team more organized, to auto-generate some of repetitive reports about the progress of the ongoing project, and provide means of communication with the team leader of the team. This project was started because companies these days are becoming bigger and bigger with more employees and more teams in the company. Due to this, the organization of teams in companies is becoming more complex. This application will be developed using Xamarin along with C# as a backend language, SQL as a database language, ASP.NET RESTful Web API as a Web Service for the application. In the end, the final output of the project is a well-designed, well documented easily accessible and easy to use application called the Team Project Tracker that will also be available on Google's Play Store.

Keywords: Team, Team Leader, Android, Mobile Application, Project Tracker, Employees

Table of Contents

ABSTRACT.....	II
LIST OF FIGURES	VI
LIST OF TABLES	IX
1. INTRODUCTION	1
2. PROJECT PLANNING AND MANAGEMENT	2
2.1. Project Team	2
2.2. Organization Scheme	3
2.3. Tools/Methods Applied.....	3
2.4. Reason for starting the Project	3
2.5. Success Criteria	4
2.6. Software Development Plan.....	4
2.7. Software Cost Estimation - COCOMO.....	5
2.8. Work Packages and Gantt Chart	6
2.9. List of Milestones.....	17
2.10. List of Risks	17
2.11. Commercialization Potential	18
2.12. Project Economic Expectations.....	18
2.13. Software Purchases	19
2.14. Quarterly Estimated Cost Form (TL)	21
3. REQUIREMENTS ANALYSIS	22
3.1. Functional Requirements.....	22
3.2. Non-Functional Requirements	25

3.2.1. Software Quality Attributes.....	25
3.2.2. Performance Requirements.....	25
3.2.3. Safety Requirements.....	25
3.2.4. Implementation Requirements.....	26
3.2.5. Security Requirements.....	26
3.3 Ethical issues	26
4. DESIGN.....	27
4.1. High level design (architectural)	27
4.1.1. Context Diagram.....	27
4.1.2. Dataflow Diagram – Level 0	28
4.1.3. User Interface Design	29
4.2. Low level design (components used)	41
4.2.1. Dataflow Diagram – Level 1	41
4.2.2. Data Dependencies	42
4.2.3. Class Diagram.....	43
4.2.4. UML Interaction Diagrams	44
5. IMPLEMENTATION.....	49
5.1. Tools, technologies and platforms used	49
5.2. Use of Software Engineering Process Steps	49
5.3. Algorithms.....	50
5.4. Standards	51
5.5. Detailed description of the implementation	52
5. TESTING.....	59

6. USER GUIDE OF THE SYSTEM	61
7. DISCUSSION	66
9. CONCLUSION.....	67
10. REFERENCES	68
APPENDIX A.....	69
APPENDIX B	70
APPENDIX C	77

LIST OF FIGURES

Fig 1. Organization Scheme.....	3
Fig 2. Gantt Chart.....	11
Fig 3. Context Diagram.....	27
Fig 4. Level 0 Data Flow Diagram.....	28
Fig 5. Login Page.....	29
Fig 6. Registration Page.....	30
Fig 7. Joined Teams – Main Page.....	31
Fig 8. Create Team Page.....	32
Fig 9. Successful Team Creation.....	33
Fig 10. Join a Team.....	34
Fig 11. Team Dashboard – Member.....	35
Fig 12. Team Dashboard – Admin.....	36
Fig 13. Send a Report.....	37
Fig 14. Messages from Admin.....	38
Fig 15. Reports sent to admin.....	39
Fig 16. Messages from Admin.....	40
Fig 17. Level 1 Data Flow Diagram (Sending a Report).....	41
Fig 18. ERD for Team Project Tracker.....	42
Fig 19. Class Diagram for Team Project Tracker.....	43
Fig 20. Use Case Diagram.....	44
Fig 21. Sequence Diagram – Login to System.....	45
Fig 22. Sequence Diagram – Register a new User.....	46

Fig 23. Activity Diagram – Create a Team.....	47
Fig 24. Business Process Model.....	48
Fig 25. Incremental Method of Development.....	49
Fig 26 (i). Database Implementation in Microsoft SQL Server Management Studio	53
Fig 26 (ii). Database Implementation in Microsoft SQL Server Management Studio.....	54
Fig 27. Database Implementation on Microsoft Azure Cloud Services.....	55
Fig 28. Visual Studio Project Files.....	56
Fig 29 (i). Implementation of ASP.NET RESTful Web Service.....	57
Fig 29 (ii). Implementation of ASP.NET RESTful Web Service.....	58
Fig 30. Login/Reg_Guide.....	62
Fig 31. Registration Page_Guide.....	62
Fig 32. Teams Dashboard_Guide.....	62
Fig. 33 Team Creation_Guide.....	62
Fig 34. Teams credential_Guide.....	63
Fig 35. Team Join_Guide.....	63
Fig 36. Member dashboard_Guide.....	63
Fig 37 (i) Send a Report_Guide.....	63
Fig 37 (ii) Send a Report_Guide.....	64
Fig 38. Announcements_Guide.....	64
Fig 39. Reports Sent_Guide.....	64
Fig 40. Admin dashboard_Guide.....	64
Fig 41. Create Announcement_Guide.....	65
Fig 42. List Members_Guide.....	65

Fig 43 (i). Report Details_Guide.....	65
Fig 43 (ii). Report Details_Guide.....	65

LIST OF TABLES

Table 1. Project Team.....	2
Table 2. UAF Table.....	5
Table 3. Work Package 1.....	6
Table 4. Work Package 2.....	7
Table 5. Work Package 3.....	8
Table 6. Work Package 4.....	9
Table 7. Work Package 5.....	10
Table 8. Work Breakdown Structure.....	16
Table 9. List of Milestones.....	17
Table 10. List of Risks.....	17
Table 11. Project Economic Expectations.....	18
Table 12. Software Purchases.....	19
Table 13. Quarterly Estimated Cost Form.....	21
Table 14. Test Case 1.....	59
Table 15. Test Case 2.....	60
Table 16. Test Case 3.....	60

1. INTRODUCTION

In this project, we should develop a fully functional open source and free to use Android Application for employees in teams called the Team Project Tracker in a period of around 4 months. To do this, we used the Incremental Model of development which is a flexible model for small-medium projects. The main benefiter from this project are Team Leaders along with team members who will find that their work progress will become organized. This application is not intended for use by a specific company or team, but any person in the world could use the application and create his own team for his own personal projects. The need for this project arises from the fact that in today's world, people are becoming more and more dependent on devices such as smartphones to organize their lives. Also, companies are becoming bigger with more and more employees and their organization of teams is becoming more complex. By doing this project, a lot of managers will use this application as their main way to record what their employees have been up to, how they are doing on their projects and to communicate with them. Before writing this report, two main reports were formed, the first one is the Software Requirements Specification Document¹ (SRS) and the second one is the Software Design Specification Document² (SDS). In this report, we will go through all the phases that lead to the building of this project. First of all, the specific requirements of the system will be described. Then, the design of various parts of the system will be shown. After that, the implementation of the system will be explained. Finally, the testing phase will be explained and a user guide will also be shown.

2. PROJECT PLANNING AND MANAGEMENT

2.1. Project Team

Table 1. Project Team

Project No	5
Project Name	Team Project Tracker
Start Date	25-Sep-2017
End Date	06-Jan-2018
Time	104 Days

Team Leader/Software Architect/Software Designer/Documenter			
Name Surname	Talal H. B. Mahdy	ID No	147139
Address	Famagusta, North Cyprus		
Phone	+90 533 8885729		
Email	talal.mahdy96@gmail.com		

Lead Programmer/Database Developer/Administrator			
Name Surname	Mohamed M. M. Balto	ID No	147697
Address	Famagusta, North Cyprus		
Phone	+90 533 8397554		
Email	Baltu.libya@gmail.com		

Software Programmer/Software Tester/Maintainer			
Name Surname	Abdoulgwad Hussien Elsheredi	ID No	147597
Address	Famagusta, North Cyprus		
Phone	+90 533 8528065		
Email	abdoulgwad.elsheredi@yahoo.it		

Requirements Engineer/User Interface Designer			
Name Surname	Adham Moshasha	ID No	148387
Address	Famagusta, North Cyprus		
Phone	+90 533 8725650		
Email	adhamoshasha@gmail.com		

2.2. Organization Scheme

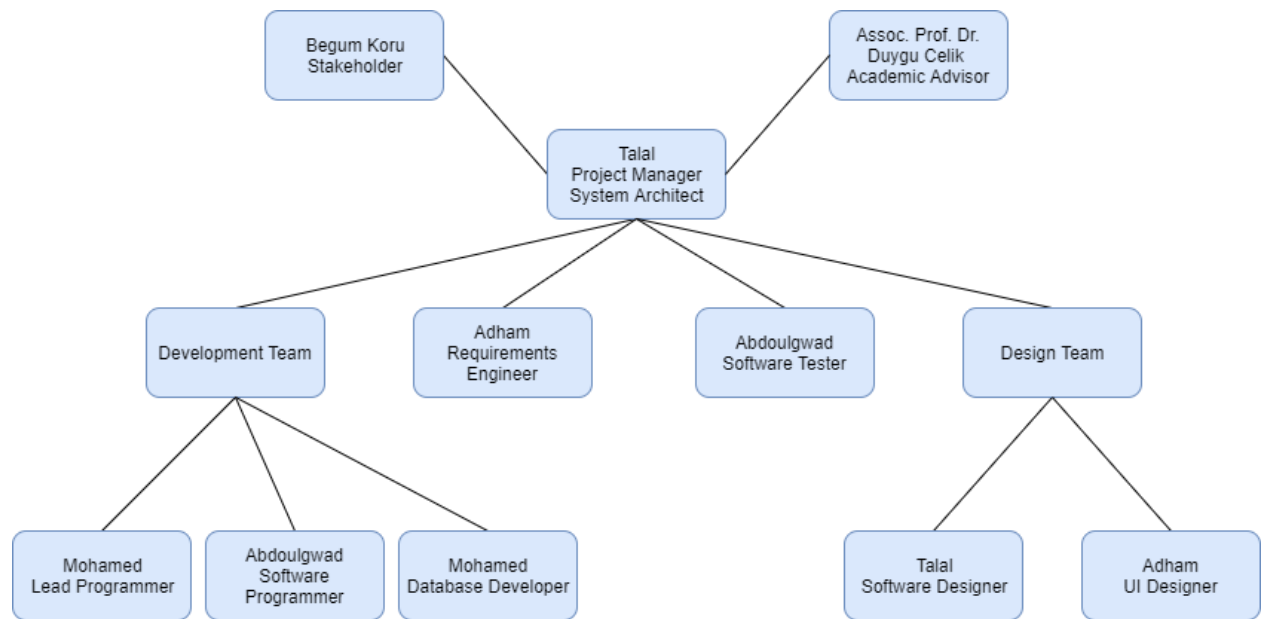


Fig 1. Organization Scheme

2.3. Tools/Methods Applied

Project Management and Scheduling: Microsoft Project³

Design: Modelio⁴, Microsoft Visio⁵, Mockflow Wireframe⁶, draw.io⁷

Implementation: Microsoft SQL Server Management Studio⁸, Visual Studio IDE⁹, Microsoft Azure¹⁰, Microsoft Xamarin⁹, C# Programming Language, JSON¹¹, ASP.NET RESTful Web API⁹

2.4. Reason for starting the Project

In these days, companies are becoming bigger with more and more employees and their organization of teams is becoming more complex. This means that there is a lot of paperwork going on inside the company to keep track of the progress that the employees in those teams are doing. Also, people these days are more and more dependent on devices such as smartphones to organize their lives and to save time. Looking at Google's Play Store, there are not very few applications like Team Project Tracker and it is feasible and makes sense to develop a project like this. After teams in companies start using the application, the company may even require its employees to use the application to record some data.

2.5. Success Criteria

Success Criteria are metrics to determine if the project is successful. Some of them are:

- a. Total Downloads: The number of times the application was downloaded should be growing at a steady rate.
 - b. Monthly Average Users (MAU): The application should have a high number of Average users among those who downloaded the application. If it appears that the MAU is growing, then the project is growing in the right path.
 - c. Engagement: Also, those users should have a high engagement ratio, i.e., users visit it frequently and use it for a considerable amount of time. Engagement can be measured by metrics such as session length (time period between app open and close), session interval (time between the user's first session and their next one) and retention rate (users who return to your app based on the date of their first visit).
 - d. Documentation: The number of users submitting help requests or bug reports should be less due to well documentation good design of application.
- 2.6. Software Development Plan

2.6. Software Development Plan

In this project, we will be applying an evolutionary development approach. An evolutionary development is based on the idea of developing an initial implementation, exposing it to the customer's comments, refining it through many versions until an adequate system has been developed. This development method is going to be more effective in this project since it is a small-medium sized system. The requirements for this system are not well defined from the beginning and we have to work with the customer and produce prototypes while obtaining feedbacks from the customer. To do this, we are going to have to conduct a number of interviews with the customer and clearly understand the requirements. The main advantages of using this approach to develop this project are:

- Each module passes through requirements, design, implementation and testing phases.
- Most important Modules are developed first.
- It is more flexible and less costly to change requirements.
- It is easier to test and debug the modules.
- Delivery of initial Modules is quick.

2.7. Software Cost Estimation - COCOMO

Function Point Calculation:

Table 2. UAF Table

			Weighting factor				
Measurement parameter	Count		Simple	Average	Complex		
Number of user inputs	22	×	3	4	6	=	132
Number of user outputs	7	×	4	5	7	=	35
Number of user inquiries	4	×	3	4	6	=	12
Number of files	6	×	7	10	15	=	60
Number of external interfaces	1	×	5	7	10	=	5
Count total						=	244

Function Points = UFP*TCF

Function Points = UFP*[0.65+0.01*DI]

Unadjusted Function Points (UFP) = 244

Technical Complexity Factor (TCF):

We assume that all Complexity Adjustment Factors are average (Value: 3 on a scale of 5).

DI = summation of all Complexity Adjustment Factors (Total 14) according to influence

Therefore, DI = 3x14=42

Therefore, Function Points = 244*[0.65+0.01*42] = 261 Function Points

FP to SLOC Conversion Ratio for C# Language = 29

Therefore, Lines of Code (LOC) = 261 * 29 = 7569 LOC = 7.569 KLOC

2.8. Work Packages and Gantt Chart

Table 3. Work Package 1

Work Package No	1
Work Package Name	Feasibility and Pre-Research (SRS stage)
Start-End Date and Time	Start: 25-09-17 Finish: 12-10-17
1- Activities of work package.	
1. Scope. 2. Analysis/Software Requirements.	
2- Methods and parameters that will be used for work package.	
None.	
3- Experiments, tests and analysis in the work package.	
1. Scope: 1.1. Determine project scope 1.2. Secure project approval 1.3. Define preliminary resources 1.4. Secure core resources 1.5. Scope complete 2. Analysis/Software Requirements: 2.1. Conduct needs analysis 2.2. Draft preliminary software specifications 2.3. Develop preliminary budget 2.4. Review software specifications/budget with team 2.5. Incorporate feedback on software specifications 2.6. Develop delivery timeline 2.7. Obtain approvals to proceed (concept, timeline, budget) 2.8. Secure required resources 2.9. Analysis complete	
4- Output of work package and its success criteria.	
Outputs: Initial Requirements Specification Document (SRS), feasibility analysis, secured resources. Success Criteria: Project approved, project is feasible to implement, initial requirements are well documented, resources and team members are secured.	
5- Relation of output with other work packages	
This is the initial phase of development and is the basic input for all other work packages. It defines the following: What is the project? Who are the stakeholders? Who will use the system? How should it be developed? Who are the team members? What are the basic requirements? How should it be developed? How should it be delivered? Etc.	

Table 4. Work Package 2

Work Package No	2
Work Package Name	System Design (SDS Stage)
Start-End Date and Time	Start: 12-10-17 Finish: 03-11-17
1- Activities of work packages.	
1. Team Project Tracker Software Design 2. Development of first prototype 3. Improve SRS Document	
2- Methods and parameters that will be used for work package.	
Visual Paradigm, Microsoft Visio, Mockflow Wireframe, Modelio, draw.io	
3- Experiments, tests and analysis in the work package.	
Review preliminary software specifications Develop functional specifications Design of System Develop prototype based on functional specifications Review functional specifications and Design Incorporate feedback into functional specifications Obtain approval to proceed Design complete	
4- Output of work package and its success criteria.	
Outputs: A Software Design Specification (SDS) Document, First Prototype of Software. Success Criteria: An improvement of the SRS Document as a result of better understanding of requirements from first prototype, completion of system design.	
5- Relation of output with other work packages	
The design stage is the next stage in the software development life cycle. Without designing the software and knowing what has to be done, it will be very difficult for the programmer to develop the software and many mistakes will be done. So this work package is a very important prerequisite to the next stage which is the development stage.	

Table 5. Work Package 3

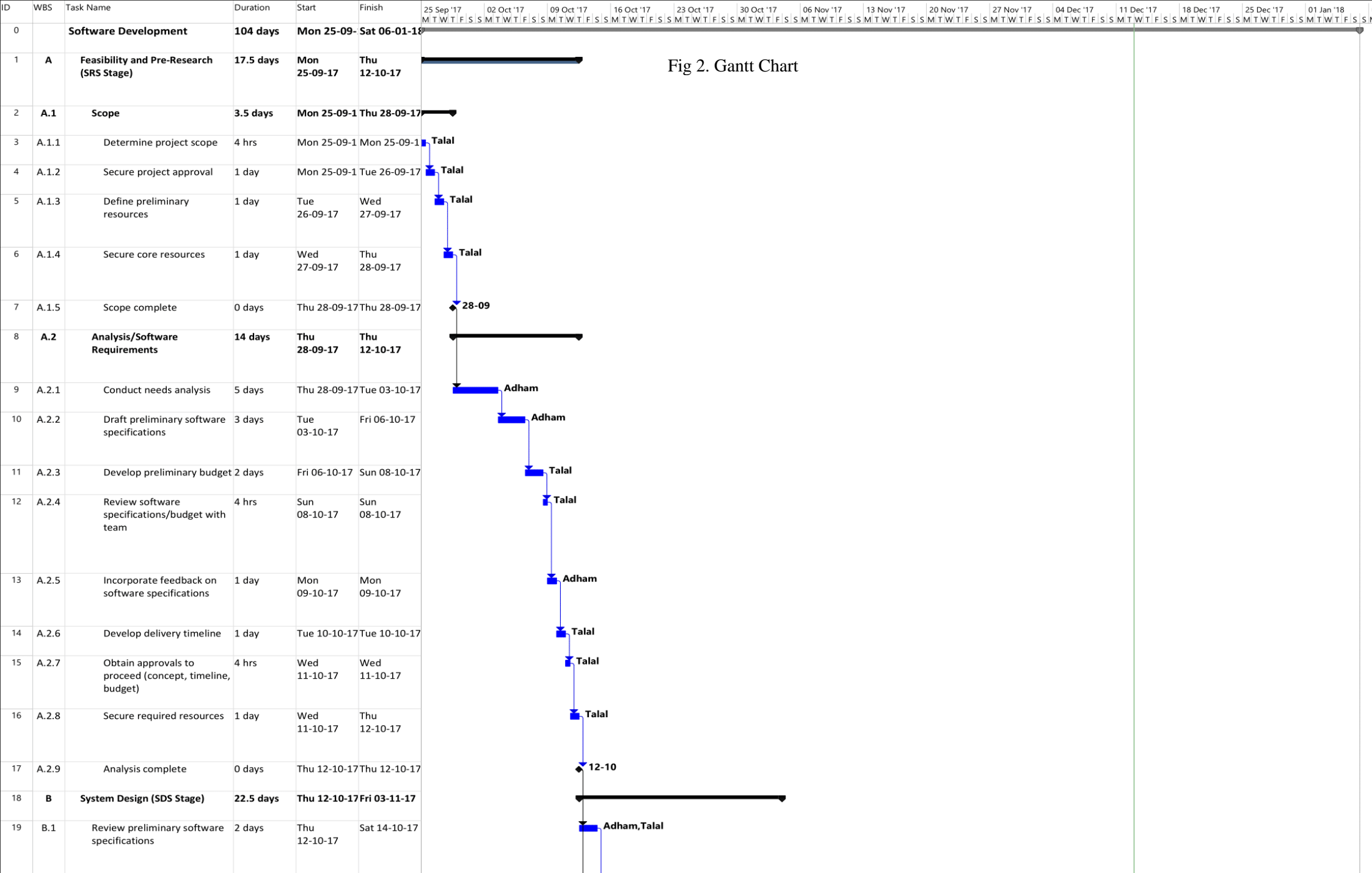
Work Package No	3
Work Package Name	Software Development Stage
Start-End Date and Time	Start: 04-11-17 Finish: 26-11-17
1- Activities of work packages.	
The main coding, primary debugging of the program and development of the database.	
2- Methods and parameters that will be used for work package.	
XAML, C#, Microsoft SQL Server Management Studio, Visual Studio IDE, Microsoft Azure, JSON, ASP.NET RESTful Web API	
3- Experiments, tests and analysis in the work package.	
Review functional specifications Identify modular/tiered design parameters Assign development staff Develop Code and Database Developer testing (primary debugging) Development complete	
4- Output of work package and its success criteria.	
Outputs: Team Project Tracker Android Application Package (APK) Success Criteria: A successful working APK file of our project.	
5- Relation of output with other work packages	
During the development of our application, the coders will obviously find some bugs and attempt to fix them. However, there might be some logical or other types of errors that a developer might not notice. Therefore, it is important for the application to be tested by a separate dedicated tester. Testing of the application can begin shortly after the development of the first unit of the application.	

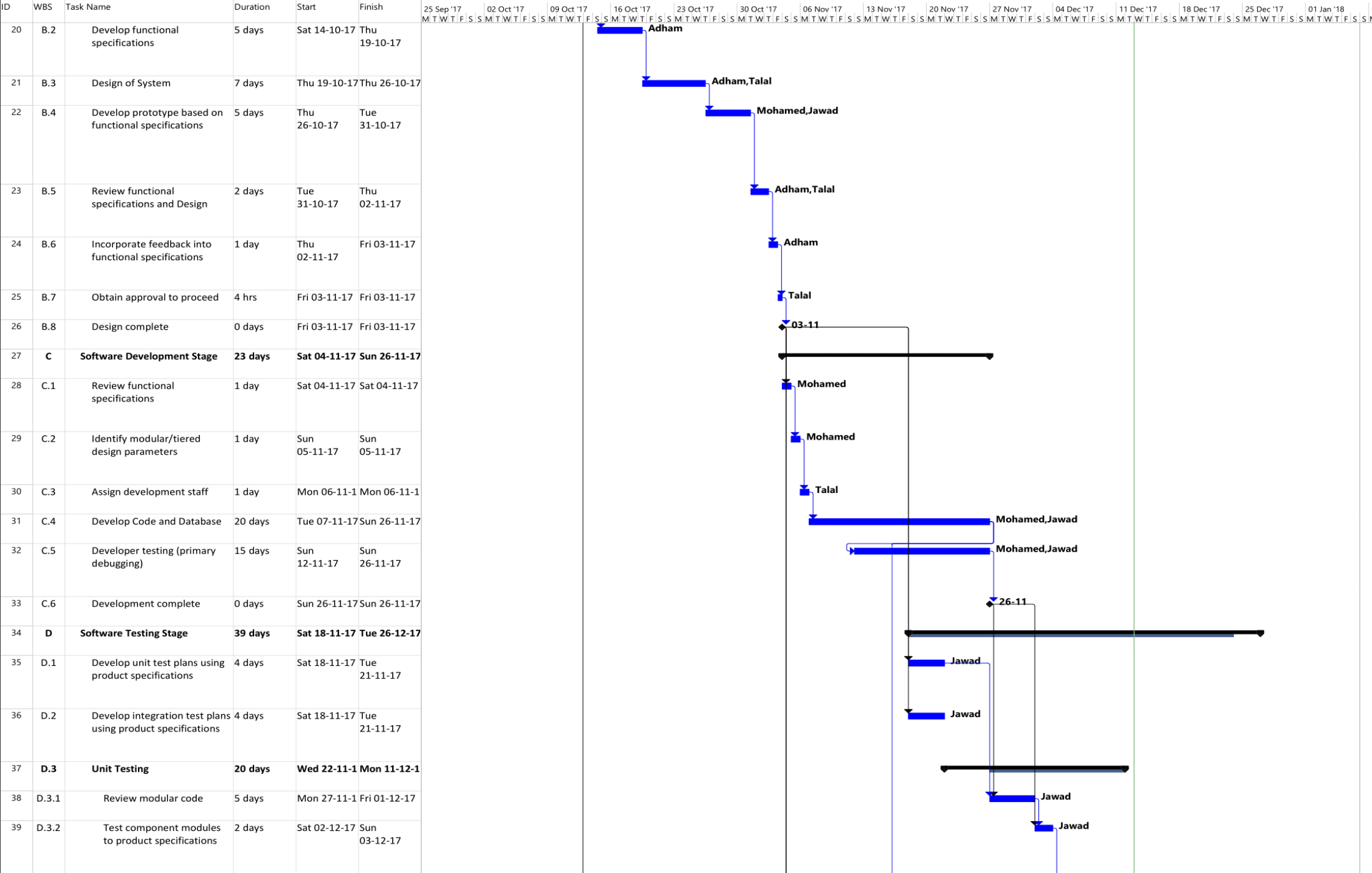
Table 6. Work Package 4






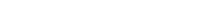
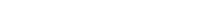
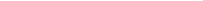
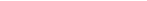






Work Package No	4
Work Package Name	Software Testing Stage
Start-End Date and Time	Start: 18-11-17 Finish: 26-12-17
1- Activities of work packages.	
<ol style="list-style-type: none"> 1. Unit and Integration Test Plans. 2. Unit Testing. 3. Integration Testing. 	
2- Methods and parameters that will be used for work package.	
Module-by-Module Unit Testing and Overall Integration Testing	
3- Experiments, tests and analysis in the work package.	
<p>1. Unit and Integration Test Plans:</p> <ol style="list-style-type: none"> 1.1. Develop unit test plans using product specifications 1.2. Develop integration test plans using product specifications <p>2. Unit Testing:</p> <ol style="list-style-type: none"> 2.1. Review modular code 2.2. Test component modules to product specifications 2.3. Identify anomalies to product specifications 2.4. Modify code 2.5. Re-test modified code 2.6. Unit testing complete <p>3. Integration Testing:</p> <ol style="list-style-type: none"> 3.1. Test module integration 3.2. Identify anomalies to specifications 3.3. Modify code 3.4. Re-test modified code 3.5. Integration testing complete 	
4- Output of work package and its success criteria.	
<p>Outputs: Test data, verification results</p> <p>Success Criteria: Testing successfully completed with all the errors and bugs successfully fixed.</p>	
5- Relation of output with other work packages	
<p>After successfully testing the system, next stages in the software life cycle are the delivery and maintenance stages. The software should be delivered and installed as per the request of the customer. Also, the maintenance stage is very important as software may serve for many years to come and it will obviously need to be updated. Therefore, a good maintenance team along with good documentation is very important for the product to be successful.</p>	

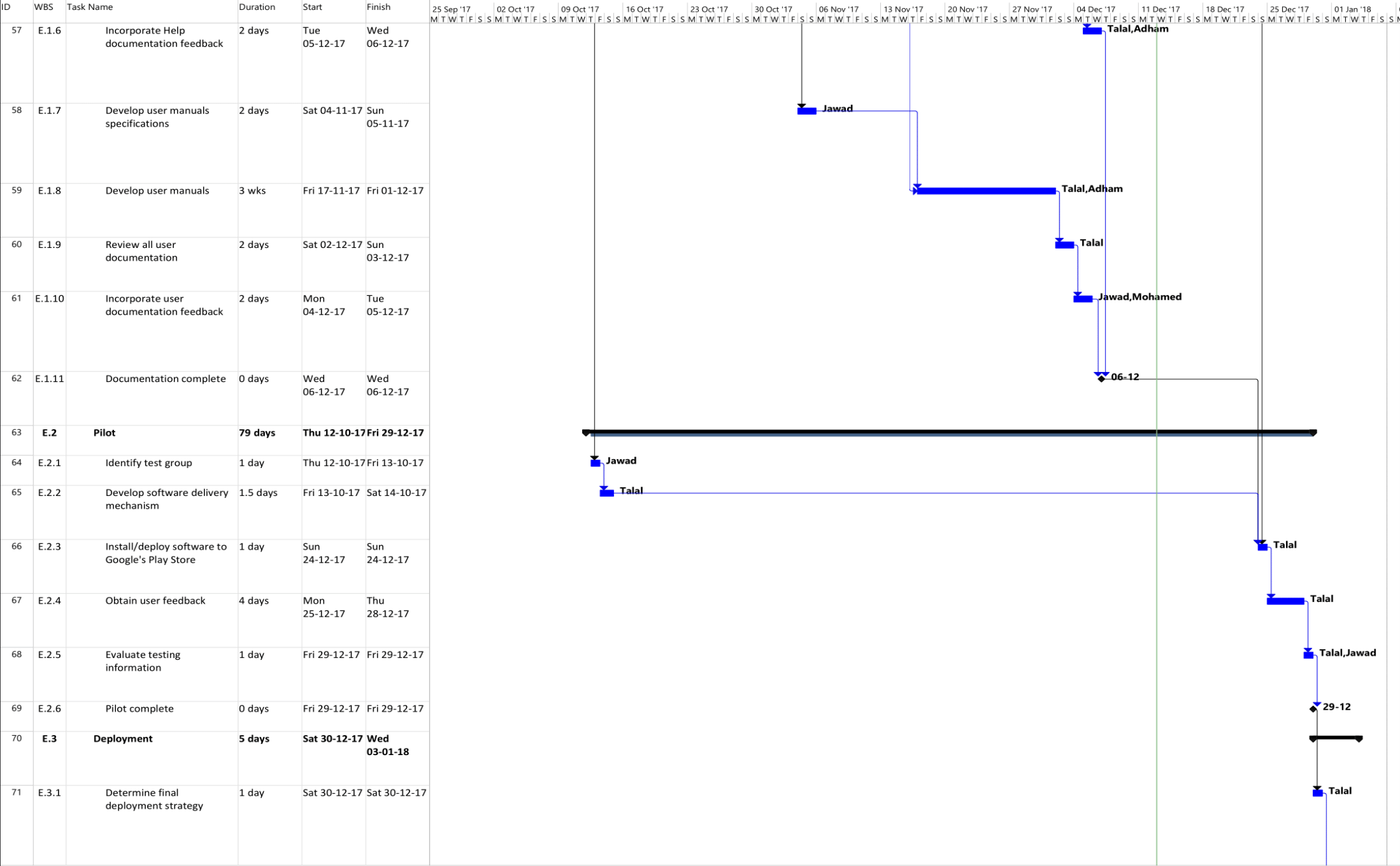
Table 7. Work Package 5

Work Package No	5
Work Package Name	Documentation and Delivery
Start-End Date and Time	Start: 12-01-18 Finish: 06/01/18
1- List the activities of work packages.	
<ol style="list-style-type: none"> 1. Documentation 2. Pilot 3. Deployment 4. Post Implementation Review 	
2- Describe the methods and parameters that will be used for work package.	
Microsoft Project, Microsoft Office, GitHub, Google Play Store	
3- List the experiments, tests and analysis in the work package.	
<ol style="list-style-type: none"> 1. Documentation <ol style="list-style-type: none"> 1.1. Develop Help specification 1.2. Develop SRS Document 1.3. Develop SDS Document 1.4. Develop Help system 1.5. Review Help documentation 1.6. Incorporate Help documentation feedback 1.7. Develop user manuals specifications 1.8. Develop user manuals 1.9. Review all user documentation 1.10. Incorporate user documentation feedback 1.11. Documentation complete 2. Pilot <ol style="list-style-type: none"> 2.1. Identify test group 2.2. Develop software delivery mechanism 2.3. Install/deploy software to Google's Play Store 2.4. Obtain user feedback 2.5. Evaluate testing information 2.6. Pilot complete 3. Deployment <ol style="list-style-type: none"> 3.1. Determine final deployment strategy 3.2. Develop deployment methodology 3.3. Secure deployment resources 3.4. Train support staff 3.5. Deploy software 3.6. Deployment complete 4. Post Implementation Review <ol style="list-style-type: none"> 4.1. Document lessons learned 4.2. Distribute to team members 4.3. Create software maintenance team 4.4. Post implementation review complete 	
4- List the output of work package and its success criteria.	
Outputs: Successful delivery of project, uploading to Play Store, completed documentation Success Criteria: A completed well documented, well perceived software application.	
5- Explain the relation of output with other work packages	
As can be noticed, the documentation stage started at an early time in the software process, sometime after the design stage started. It is important to document all requirements and design aspects of the project along with a proper user guide before delivering the application.	





Project: Software Development Date: Wed 13-12-17	Task		Project Summary		Rolled Up Progress		Inactive Milestone		Manual Summary Rollup		Deadline	
	Split		Group By Summary		External Tasks		Inactive Summary		Manual Summary		Progress	
	Milestone		Rolled Up Task		External Milestone		Manual Task		Start-only			
	Summary		Rolled Up Milestone		Inactive Task		Duration-only		Finish-only			



Project: Software Development
Date: Wed 13-12-17

Task

Split

Milestone

Summary

Project Summary

Group By Summary

Rolled Up Task

Rolled Up Milestone

Rolled Up Progress

External Tasks

External Milestone

Inactive Task

Inactive Milestone

Inactive Summary

Manual Summary

Duration-only

Manual Summary Rollup

Manual Summary

Start-only

Finish-only

Deadline

Progress

Page 4

WBS	Task Name	Duration	Start	Finish	25 Sep '17	02 Oct '17	09 Oct '17	16 Oct '17	23 Oct '17	30 Oct '17	06 Nov '17	13 Nov '17	20 Nov '17	27 Nov '17	04 Dec '17	11 Dec '17	18 Dec '17	25 Dec '17	01 Jan '18
72	E.3.2	Develop deployment methodology	1 day	Sun 31-12-17	Sun 31-12-17														
73	E.3.3	Secure deployment resources	1 day	Mon 01-01-18	Mon 01-01-18														
74	E.3.4	Train support staff	1 day	Tue 02-01-18	Tue 02-01-18														
75	E.3.5	Deploy software	1 day	Wed 03-01-18	Wed 03-01-18														
76	E.3.6	Deployment complete	0 days	Wed 03-01-18	Wed 03-01-18														
77	E.4	Post Implementation Review	3 days	Thu 04-01-18	Sat 06-01-18														
78	E.4.1	Document lessons learned	1 day	Thu 04-01-18	Thu 04-01-18														
79	E.4.2	Distribute to team members	1 day	Fri 05-01-18	Fri 05-01-18														
80	E.4.3	Create software maintenance team	1 day	Sat 06-01-18	Sat 06-01-18														
81	E.4.4	Post implementation review complete	0 days	Sat 06-01-18	Sat 06-01-18														
82	F	Software development complete	0 days	Sat 06-01-18	Sat 06-01-18														

Project: Software Development Date: Wed 13-12-17	Task	<div></div>	Project Summary	<div></div>	Rolled Up Progress	<div></div>	Inactive Milestone	<div></div>	Manual Summary Rollup	<div></div>	Deadline	<div></div>
	Split	<div></div>	Group By Summary	<div></div>	External Tasks	<div></div>	Inactive Summary	<div></div>	Manual Summary	<div></div>	Progress	<div></div>
	Milestone	<div></div>	Rolled Up Task	<div></div>	External Milestone	<div></div>	Manual Task	<div></div>	Start-only	<div></div>		
	Summary	<div></div>	Rolled Up Milestone	<div></div>	Inactive Task	<div></div>	Duration-only	<div></div>	Finish-only	<div></div>		

Table 8. WBS

Team Project Tracker - Work Breakdown Structure (WBS)

					2017				18
WP NO	WORK STEPS	TIME	START DATE	END DATE	September	October	November	December	January
WP1:	Feasibility and Pre-Research (SRS stage)	18 Days	25-09-17	12-10-17					
1.1.	Scope								
	Determine project scope								
	Secure project approval								
	Define preliminary resources								
	Secure core resources								
	Scope complete								
1.2.	Analysis/Software Requirements								
	Conduct needs analysis								
	Draft preliminary software specifications								
	Develop preliminary budget								
	Review software specifications/budget with team								
	Incorporate feedback on software specifications								
	Develop delivery timeline								
WP2:	System Design (SDS Stage)	23 Days	12-10-17	03-11-17					
2.1.	Review preliminary software specifications								
2.2.	Develop functional specifications								
2.3.	Design of System								
2.4.	Develop prototype based on functional specifications								
2.5.	Review functional specifications and Design								
2.6.	Incorporate feedback into functional specifications								
2.7.	Obtain approval to proceed								
2.8.	Design complete								
WP3:	Software Development Stage	23 Days	04-11-17	26-11-17					
3.1.	Review functional specifications								
3.2.	Identify modular/tiered design parameters								
3.3.	Assign development staff								
3.4.	Develop Code and Database								
3.5.	Developer testing (primary debugging)								
3.6.	Development complete								
WP4:	Software Testing Stage	39 Days	18-11-17	26-12-17					
4.1.	Develop unit test plans using product specifications								
4.2.	Develop integration test plans using product specifications								
4.3.	Unit Testing								
	Review modular code								
	Test component modules to product specifications								
	Identify anomalies to product specifications								
	Modify code								
	Re-test modified code								
	Unit testing complete								
4.4.	Integration Testing								
	Test module integration								
	Identify anomalies to specifications								
	Modify code								
	Re-test modified code								
	Integration testing complete								
WP5:	Documentation and Delivery	87 Days	12-10-17	06-01-18					
5.1.	Documentation								
	Develop Help specification								
	Develop SRS Document								
	Develop SDS Document								
	Develop Help system								
	Review Help documentation								
	Incorporate Help documentation feedback								
	Develop user manuals specifications								
	Develop user manuals								
	Review all user documentation								
	Incorporate user documentation feedback								
	Documentation complete								
5.2.	Pilot								
	Identify test group								
	Develop software delivery mechanism								
	Install/deploy software to Google's Play Store								
	Obtain user feedback								
	Evaluate testing information								
	Pilot complete								
5.3.	Deployment								
	Determine final deployment strategy								
	Develop deployment methodology								
	Secure deployment resources								
	Train support staff								
	Deploy software								
	Deployment complete								
5.4.	Post Implementation Review								
	Document lessons learned								
	Distribute to team members								
	Create software maintenance team								
	Post implementation review complete								
	Software development complete								

2.9. List of Milestones

Table 9. List of Milestones

No.	Description of Output	Expected Time Interval		
1	Scope determination and approval	25-09-17	28-09-17	
2	Analysis/Software Requirements	28-09-17	12-10-17	
3	System Design (SDS Stage)	12-10-17	03-11-17	
4	Software Development Stage	04-11-17	26-11-17	
5	Unit Testing	22-11-17	11-12-17	
6	Integration Testing	11-12-17	23-12-17	
7	Documentation	20-10-17	06-12-17	
8	Pilot	12-10-17	29-12-17	
9	Deployment	30-12-17	03-01-18	
10	Post Implementation Review	04-01-18	06-01-18	
11	Software Development	25-09-17	06-01-18	

2.10. List of Risks

Table 10. List of Risks

Risk	Probability	Effects	Strategy
The time required to develop the software is underestimated.	High	Serious	The most important requirements of the project should always be implemented first. We will have more time later on to implement the non-important requirements.
Software tools cannot work together in an integrated way.	High	Tolerable	Always minimize the number of design tools used and make sure that the outputs of these tools are compatible with each other.
Customers fail to understand the impact of requirements changes.	Moderate	Tolerable	Conduct frequent meetings with the stakeholders and keep being updated on latest requirement changes.
The rate of defect repair is underestimated.	Moderate	Tolerable	Replace potentially defective components with more reliable bought-in components.
The size of the software is underestimated.	Moderate	Insignificant	Investigate buying software components; Investigate use of a program generator.
Code generated by code generation tools is inefficient.	Moderate	Insignificant	Risk is always expected since code generation tools often can't produce reliable code . This code always needs editing by the software developers.
Key staffs are ill at critical times in the project.	Moderate	Serious	Reorganize team so that there is more overlap of work and people therefore understand each other's jobs.
The database used in the system cannot process as many transactions per second as expected.	Low	Serious	Investigate the possibility of buying a higher-performance database.

2.11. Commercialization Potential

Commercialization of our product can start as soon as the development and testing of the most important modules is done. However, to further guarantee that everything is working as planned and to have an advantage over other applications, we are going to wait until most of the features of the application are done. After that, the commercialization process starts when the application is uploaded to Google's Play Store. Then, many user feedbacks will be gained and improvements with new features and bug fixes will be implemented. In addition, during this time, an advertisement campaign will be made to promote our product and increase the user base. So the aim is to enable and invite many companies to use this application.

2.12. Project Economic Expectations

Table 11. Project Economic Expectations

Time-to-market (month):	4
The expected increase in sales revenue (%):	25%
The expected increase in market share (%):	5%
Time to start to gain:	February 2018

2.13. Software Purchases

Table 12. Software Purchases

Project Name										
Line no	Instrument / Equipment / Software / Publication Name	No. of Item	Capacity	Technical specification	Purpose of Project Activities	Post-Project Place of Use / Purpose		Unit Price (USD)	Unit Price (TL)	Total Amount (TL)
						R & D	Production			
1	Visual Studio	1		Integrated Development Environment (IDE) from Microsoft	Main IDE used for development of our project		Yes	-	-	-
2	Microsoft Project	1		Project Management Software	We will use this application to plan and schedule our project		Yes	589.99	2085.36	-
3	Microsoft Office	1		An office suite of applications, servers, and services	Used in many areas of the project such as documentation		Yes	399.99	1413.80	-

4	Microsoft Visio	1		Software Design Tool	Used to draw software design diagrams		Yes	299.99	1060.35	-
5	Visual Paradigm	1		Software Design Tool and Code generator	Used to draw software design diagrams and generate code required for the application based on those diagrams		Yes	349	1233.54	-
6	Mockflow Wireframe	1		User Interface Design Tool	Used to draw a User Interface for our system		Yes	208	735.38	-
7	Modelio	1		Software Design Tool and Code generator	Used to draw software design diagrams and generate code required for the application based on those diagrams		Yes	-	-	-
									TOTAL	6528.4 TL

2.14. Quarterly Estimated Cost Form (TL)

Table 13. Quarterly Estimated Cost Form

Project Name :				
Cost Item	2017		TOTAL (TL)	TOTAL COST RATE OF CONTENTS (%)
	I	II		
Personnel	20000	20000	40000	54.75
Travel	1000	1000	2000	2.73
Software Costs	6528.4	-	6528.4	8.93
Research, Development and Testing Costs	3100	3100	6200	8.48
Domestic Deployment Services and Maintenance Procurement Cost	1800	1800	3600	4.92
Overseas Deployment Services and Maintenance Procurement Cost	2000	2000	4000	5.47
Research Material	2100	2100	4200	5.74
TOTAL COST	36528.4	36528	73056.4	On a scale of <u>100</u>
IN THE PROJECT TOTAL MAN-MONTH			720 hours	

3. REQUIREMENTS ANALYSIS

3.1. Functional Requirements

Login:

- REQ-1: The user should be able to input his username in the username text box.
- REQ-2: The user should be able to input his password in the password text box.
- REQ-3: The user should be able to click on the “login” button.
- REQ-4: The system should check if there are errors in the username/password entered and generate an error message if errors exist.
- REQ-5: The system should load up the home page of the specific user after authenticating his username/password.
- REQ-6: The user should be able to click on “New User?” button.
- REQ-7: “New User?” should navigate to the registration page once clicked on.
- REQ-8: The user should be able to click on “Forgot Your Password?” button
- REQ-9: “Forgot Your Password?” should navigate to the password recovery page once clicked on.
- REQ-10: The user should be able to click on “Help” button.
- REQ-11: “Help” should navigate to the Help page once clicked on.
- REQ-12: The user should be able to click on “About Us” button.
- REQ-13: “About Us” should navigate to the About Us page once clicked on.

Register new user:

- REQ-14: The user should be able to enter his name (text only) in the text-box next to Name.
- REQ-15: The user should be able to enter his surname (text only) in the text-box next to Surname.
- REQ-16: The user should be able to enter his email (email authentication required) in the text-box next to email.
- REQ-17: The user should be able to enter his email (email authentication required and should match the one in REQ-3) in the text-box next to retype email.
- REQ-18: The user should be able to enter his password (password authentication required) in the text-box next to password.
- REQ-19: The user should be able to enter his password (password authentication required and should match the one in REQ-5) in the text-box next to retype password.
- REQ-20: “Register” button should create an account for the user with data entered.
- REQ-21: “Already Registered” should navigate the user back to the Log-in page when clicked on.

Recover Password:

- REQ-22: The user should be able to click on “Forgot Your Password?” button
- REQ-23: “Forgot Your Password?” should navigate to the password recovery page once clicked on.

Help and About Us:

- REQ-24: The user should be able to click on “Help” button.
- REQ-25: “Help” should navigate to the Help page once clicked on.
- REQ-26: The user should be able to click on “About Us” button.
- REQ-27: “About Us” should navigate to the About Us page once clicked on.

Joined Teams Page:

- REQ-28: The user should be able to view and click on his joined teams as a list.
- REQ-29: The user should be able to create a team using the “create team” button.
- REQ-30: The user should be able to join a team using the “join team” button.
- REQ-31: “Logout” button should log the user out the system and terminate his current session.
- REQ-32: The “create team” button. Should navigate to the Create a team page.
- REQ-33: The “join team” button. Should navigate to the Join a team page.

Create a Team:

- REQ-34: The user should be able to enter the team’s name (text only) in the text-box next to Team Name.
- REQ-35: The user should be able to enter Team password (password authentication required) in the text-box next to password.
- REQ-36: The user should be able to enter Team password (password authentication required and should match the one in REQ-2) in the text-box next to retype password.
- REQ-37: “Create Team” button should create a team with data entered and the user should be automatically join and be assigned as the admin of the team.
- REQ-38: The system should generate a TeamID and Password randomly and show it to the user.

Join a Team:

- REQ-39: The user should be able to enter the TeamID (ID generated by system) in the text-box next to TeamID.
- REQ-40: The user should be able to enter Team password (password authentication required) in the text-box next to “Password”.
- REQ-41: “Join Team” button should join a team with data entered and the user should be able to see the joined team in the Joined Teams page.
- REQ-42: The “Return to Teams” button should navigate to the joined teams page when clicked on.

Team member landing page:

- REQ-43: The “Teams Page” should show the company’s title.

- REQ-44: The “Teams Page” should show the User’s name.
- REQ-45: The “Teams Page” should show the Member’s title.
- REQ-46: The “Send a Report” button should navigate to the Send Report page when clicked on.
- REQ-47: The “Messages from Admin” button should navigate to the Messages from Admins page when clicked on.
- REQ-48: The “Team Announcements” button should navigate to the Announcements page when clicked on.
- REQ-49: The “Reports Sent” button should navigate to the Send Report page when clicked on.
- REQ-50: The “Leave Team” button should delete the user from the team and update the database accordingly.
- REQ-51: The “Change Title” button should let the user edit his Member’s Title and update the database accordingly.
- REQ-52: The “Return to Teams Page” button should navigate to the Joined Teams page when clicked on.

Team admin landing page:

- REQ-53: The “Managed Teams Page” should show the company’s title.
- REQ-54: The “Managed Teams Page” should show the User’s name.
- REQ-55: The “Managed Teams Page” should show the Member’s title.
- REQ-56: The “View Reports” button should navigate to the Received Reports page when clicked on.
- REQ-57: The “Send a Message” button should navigate to the Send Report page when clicked on.
- REQ-58: The “New Announcement” button should navigate to the New Announcement page when clicked on.
- REQ-59: The “View/Edit Members” button should navigate to the Members List page when clicked on.
- REQ-60: The “Delete Team” button should delete the team from the company and update the database accordingly.
- REQ-61: The “Change Title” button should let the user edit his Member’s Title and update the database accordingly.
- REQ-62: The “Return to Teams Page” button should navigate to the Joined Teams page when clicked on.

Send Report:

- REQ-63: The user should be able to see a calendar of the month
- REQ-64: The user should be able to click on the current day or the day before (if null).
- REQ-65: The user should be able to enter the amount of hours worked on a specific task on the day selected in REQ-2
- REQ-66: The user should be able to select the desired task name from a drop down menu.
- REQ-67: The user should be able to write alphanumeric characteristics in the text box next to Description of Task.
- REQ-68: The “Reset” button should reset the selections and delete the text written from the interface without sending or updating the DB
- REQ-69: The “Send Report” button should send the report to the Admin managing the team member and update the DB accordingly.
- REQ-70: The “Return to Teams Page” button should navigate to the Joined Teams page when clicked on.

Messages from Admin:

- REQ-71: The user should be able to see the team member's name at the top.
- REQ-72: A list of messages sent by the Admin should be shown on the screen.
- REQ-73: Each message can be clicked and the application will navigate to the message log.
- REQ-74: interface should show the date and the time when the message was sent.

Reports sent to Admin:

- REQ-75: The user should be able to see the team member's name at the top.
- REQ-76: A list of reports sent by the Team Member should be shown on the screen.
- REQ-77: Each report can be clicked and the application will navigate to the report and show the report.
- REQ-78: The "Return to Teams Page" button should navigate to the Joined Teams page when clicked on.

Edit/View Users:

- REQ-79: The user should be able to see a list of members in the selected team.
- REQ-80: The team member's info should show the members name and the members title alongside the edit and delete icons.
- REQ-81: The "Edit User Title" is a button and when clicked, the applications shows a drop down menu with the titles to choose from.
- REQ-82: The "Remove Member" is a button and when clicked, the applications deletes the member from the list as well as deletes the member from the team and updates the DB accordingly.
- REQ-83: The "Return to Teams Dashboard" button should navigate to the Joined Teams page when clicked on.

3.2. Non-Functional Requirements

3.2.1. Software Quality Attributes

Adaptability, availability, correctness, flexibility, maintainability, portability, reliability, reusability, robustness, testability, and usability.

3.2.2. Performance Requirements

The application shall use the performance requirements of Android Application (APK) standards for Android API Version 19 and above.

3.2.3. Safety Requirements

The application should not contain any safety threats, malware or adware to users.

3.2.4. Implementation Requirements

The system should be implemented using the Java Programming language for Android 4.4 OS (Android Kitkat).

3.2.5. Security Requirements

The system shall be a very secure system and it should implement the SHA-1 Hashing Standard as set out by the U.S. National Security Agency (NSA) in 1995 to secure the passwords within the database of the application. Only the people authorized to access the system shall be able to access it.

- SHA-1 Hashing.
- HTTPS Database encryption
- Application Login PIN.
- Secure & encrypted APK file

3.3 Ethical issues

The user of this system may send some sensitive information such as his/her password, bank account details, etc in the report. Another unauthorized user may access this sensitive information. That's why the application is more secure by implementing a Web Service as an intermediary between the Application and the Online database.

4. DESIGN

4.1. High level design (architectural)

4.1.1. Context Diagram

A context diagram is considered the highest level of design in Dataflow diagrams. The Context diagram of Team Project Tracker is shown in Figure 5.

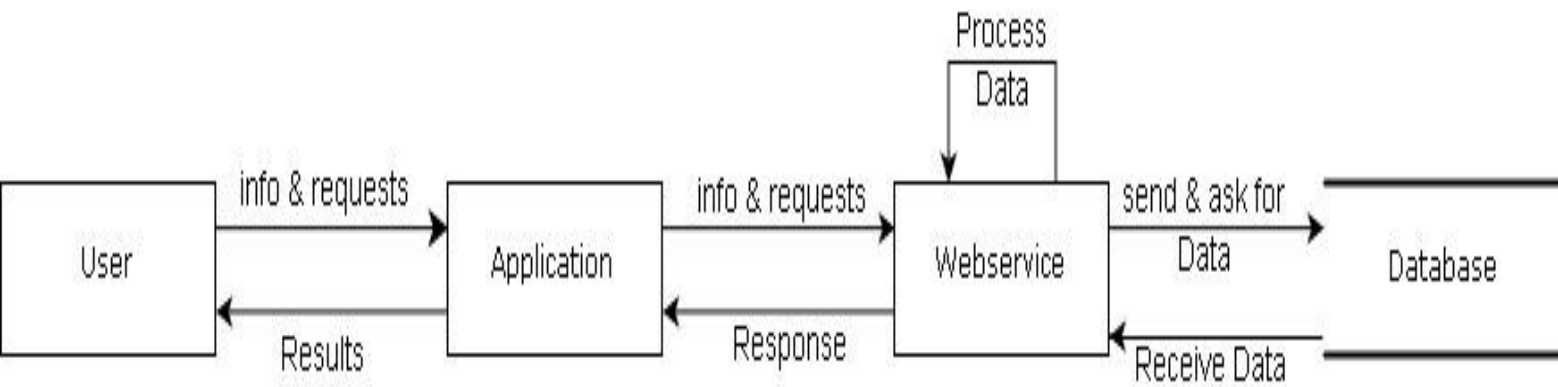


Fig 3. Context Diagram

4.1.2. Dataflow Diagram – Level 0

The second highest level in Dataflow diagram designs is a Level 0 diagram. The Level 0 diagram for Team Project Tracker is shown in Figure 6.

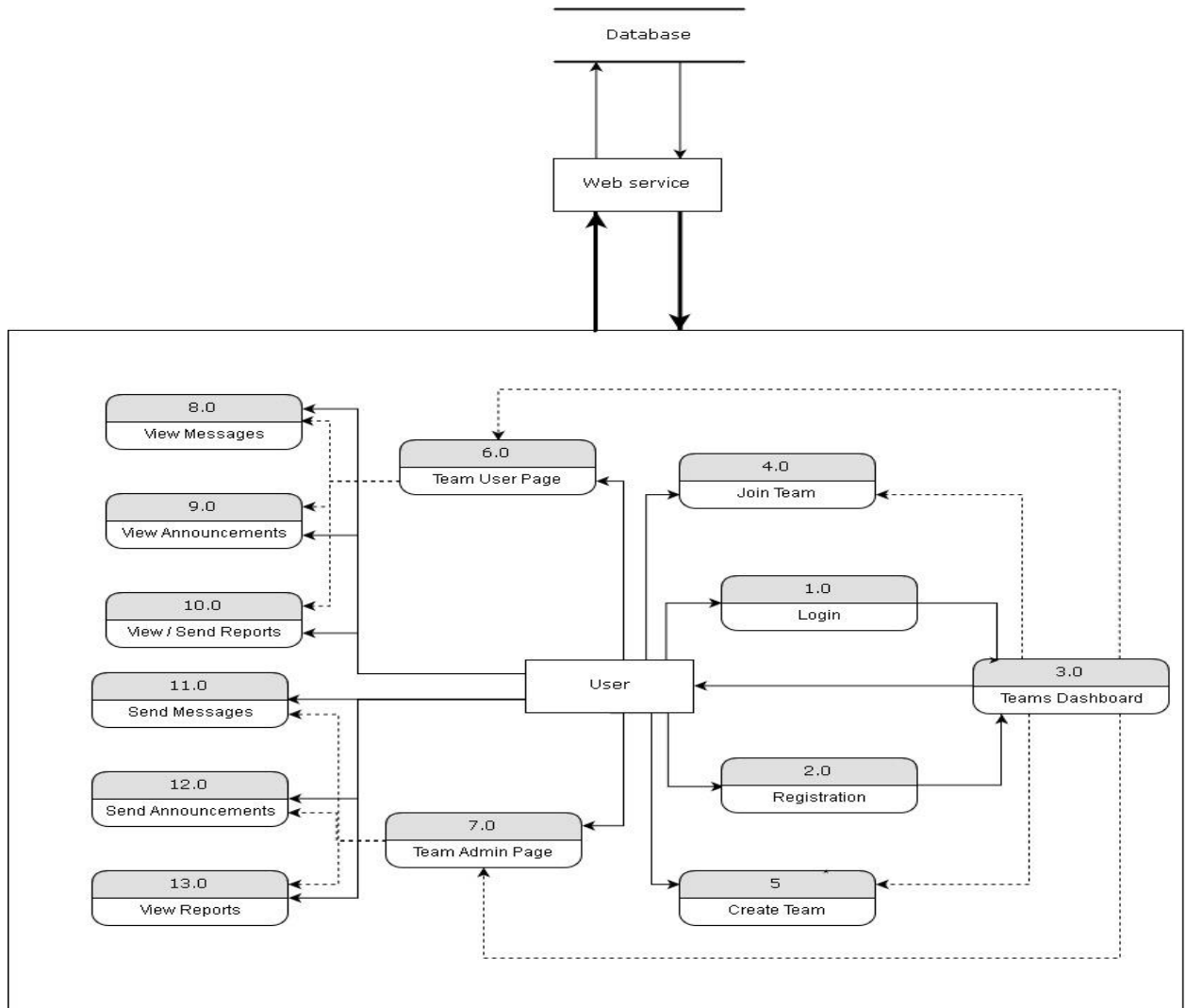


Fig 4. Level 0 Data Flow Diagram

4.1.3. User Interface Design

Each user interface for Team Project Tracker will be explained along with some screenshots in this Section. These user interfaces were created using the Mockflow Wireframe Tool.

4.1.3.1. Login

When the user first runs the application, this page will be shown. The user must register or login to access the system. The SHA-1 hashing Standard will be implemented to secure the password of the user. This is shown in Figure 8.

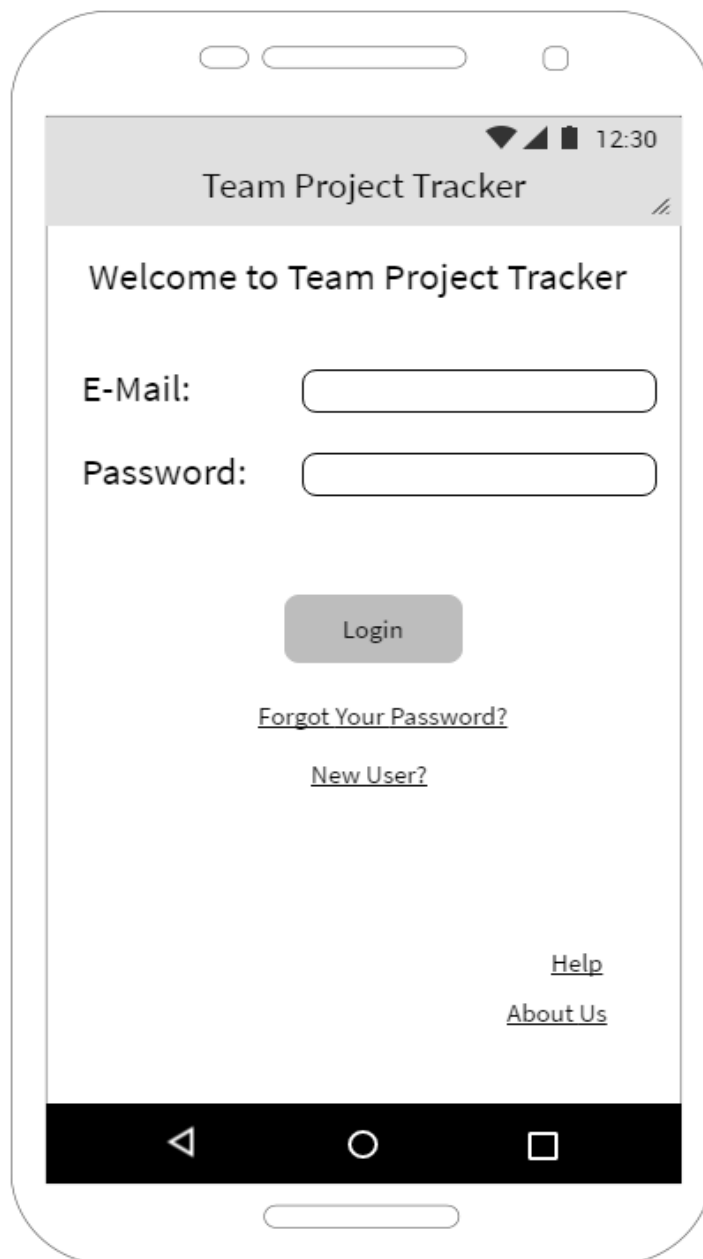
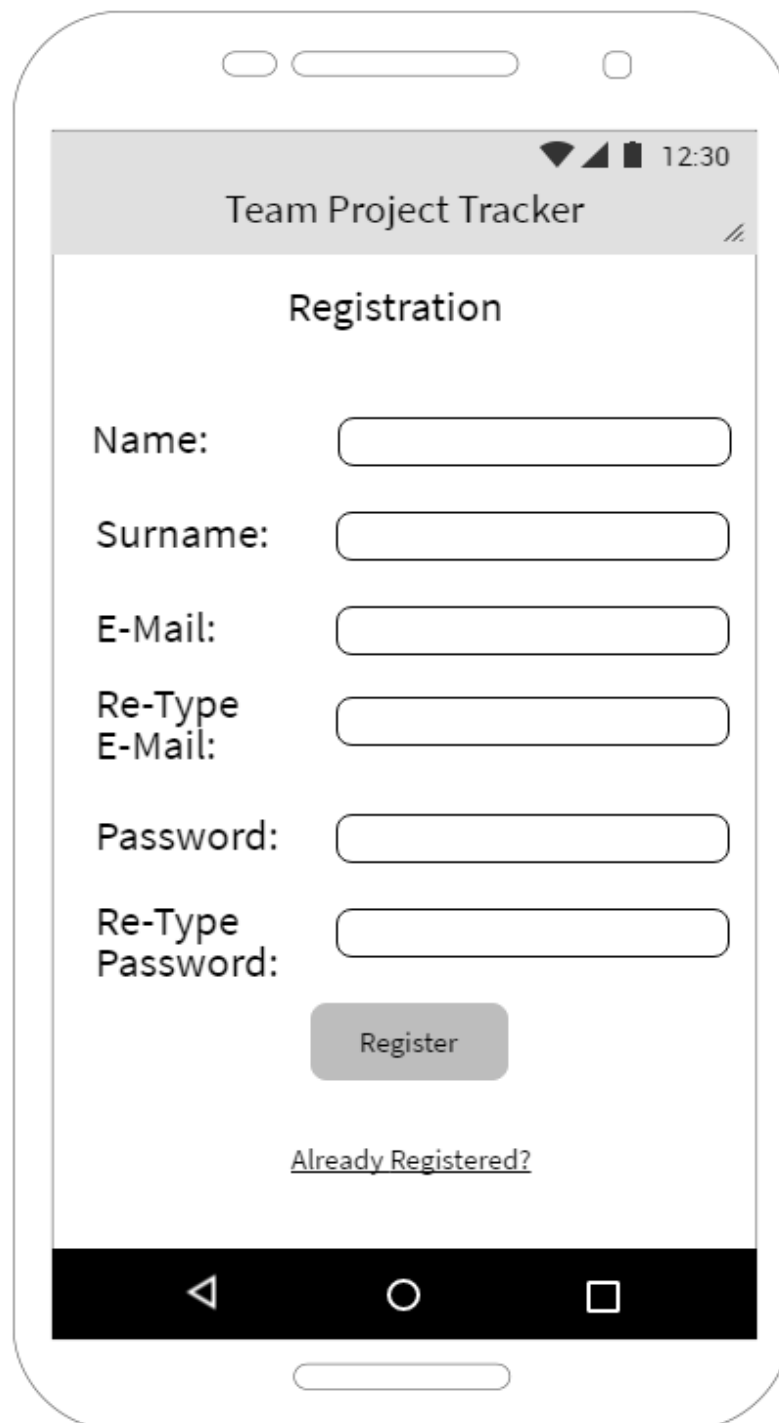


Fig 5. Login Page

4.1.3.2. Registration Page



The image shows a mobile application interface for a registration page. At the top, a grey header bar contains the text "Team Project Tracker" and a double-slash icon. Below the header, the title "Registration" is centered. The form consists of several input fields: "Name:", "Surname:", "E-Mail:", "Re-Type E-Mail:", "Password:", and "Re-Type Password:". Each label is followed by a white rectangular input box with a thin black border. Below the input fields is a grey button labeled "Register". At the bottom of the form, there is a link labeled "Already Registered?". The entire form is displayed on a smartphone screen, which has a black navigation bar at the bottom with three white icons: a triangle, a circle, and a square. The status bar at the top of the phone shows the time "12:30" and various system icons.

Team Project Tracker

Registration

Name:

Surname:

E-Mail:

Re-Type E-Mail:

Password:

Re-Type Password:

[Already Registered?](#)

Fig 6. Registration Page

4.1.3.3 Joined Teams – Main Page

The default page shown directly after logging in.

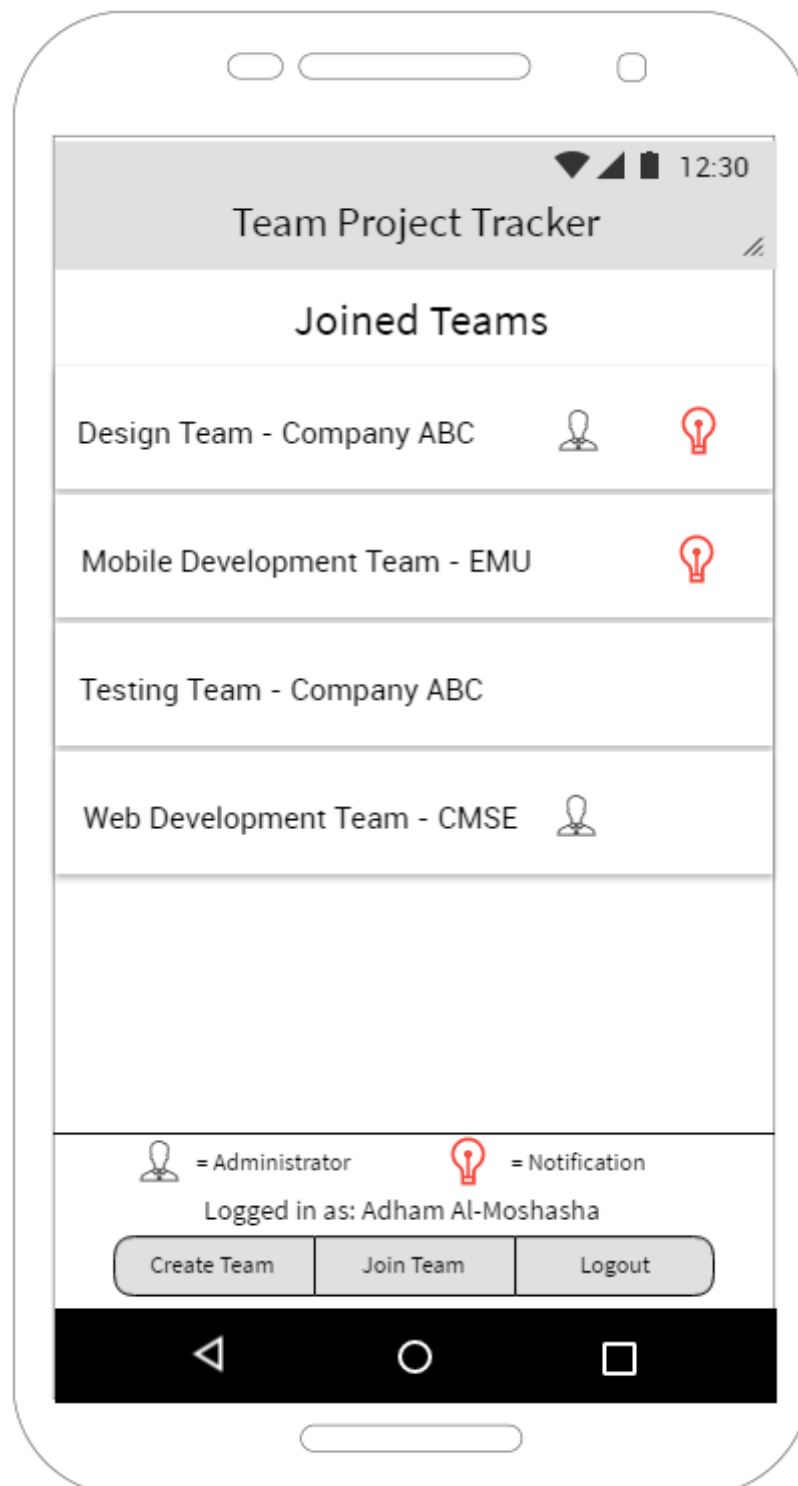


Fig 7. Joined Teams – Main Page

4.1.3.4. Create a Team

The image shows a mobile application interface for 'Team Project Tracker'. The screen is titled 'Create a Team'. It features three input fields: 'Team Name:', 'Password:', and 'Re-Type Password:'. Below these fields is a grey button labeled 'Create Team'. At the bottom of the form area is a link that says 'Return to Teams Page'. The top of the screen shows a status bar with signal, battery, and time (12:30) icons. The bottom of the screen shows a standard Android navigation bar with back, home, and recent apps buttons.

Team Project Tracker

Create a Team

Team Name:

Password:

Re-Type Password:

Create Team

[Return to Teams Page](#)

Fig 8. Create Team Page

4.1.3.5. Successful Team Creation

After a team is created, this page will be shown giving the user a TeamID. The TeamID is given since two or more teams could have the same name. The TeamID is unique and secure.

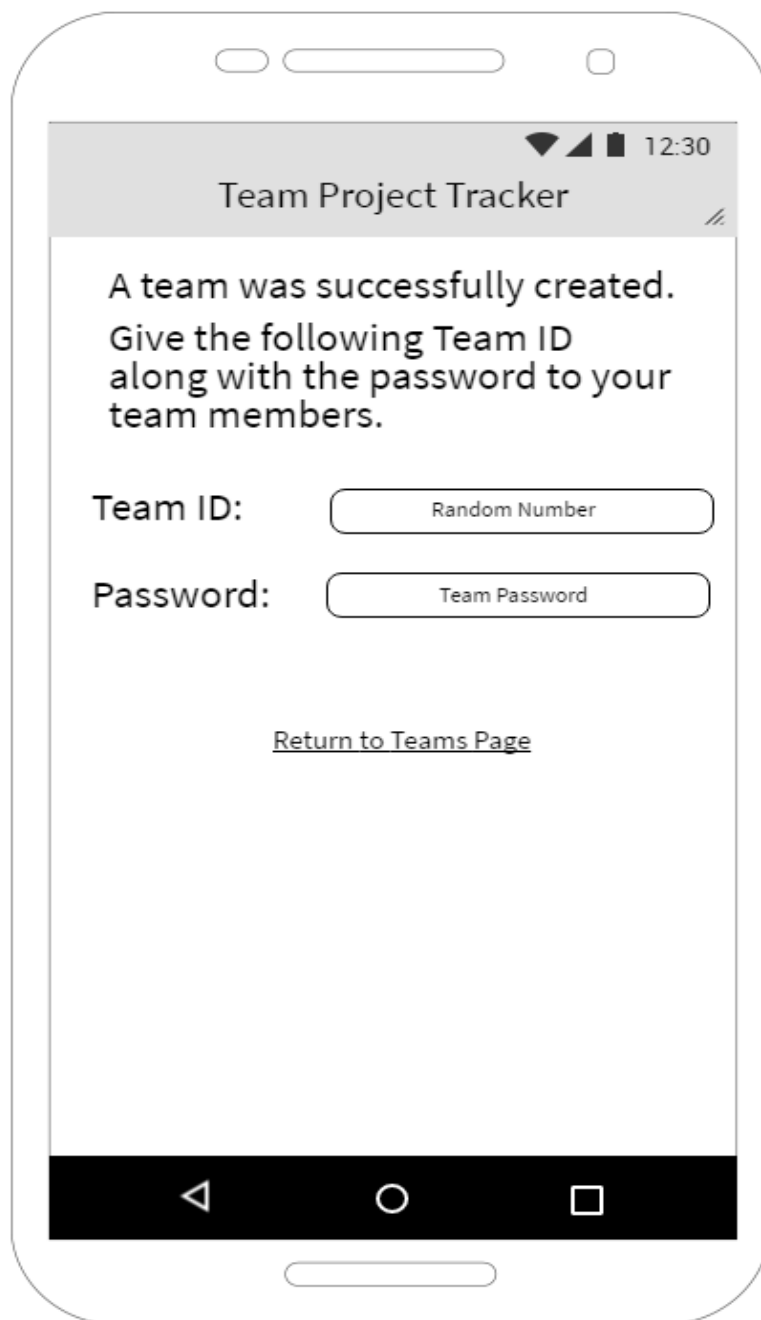
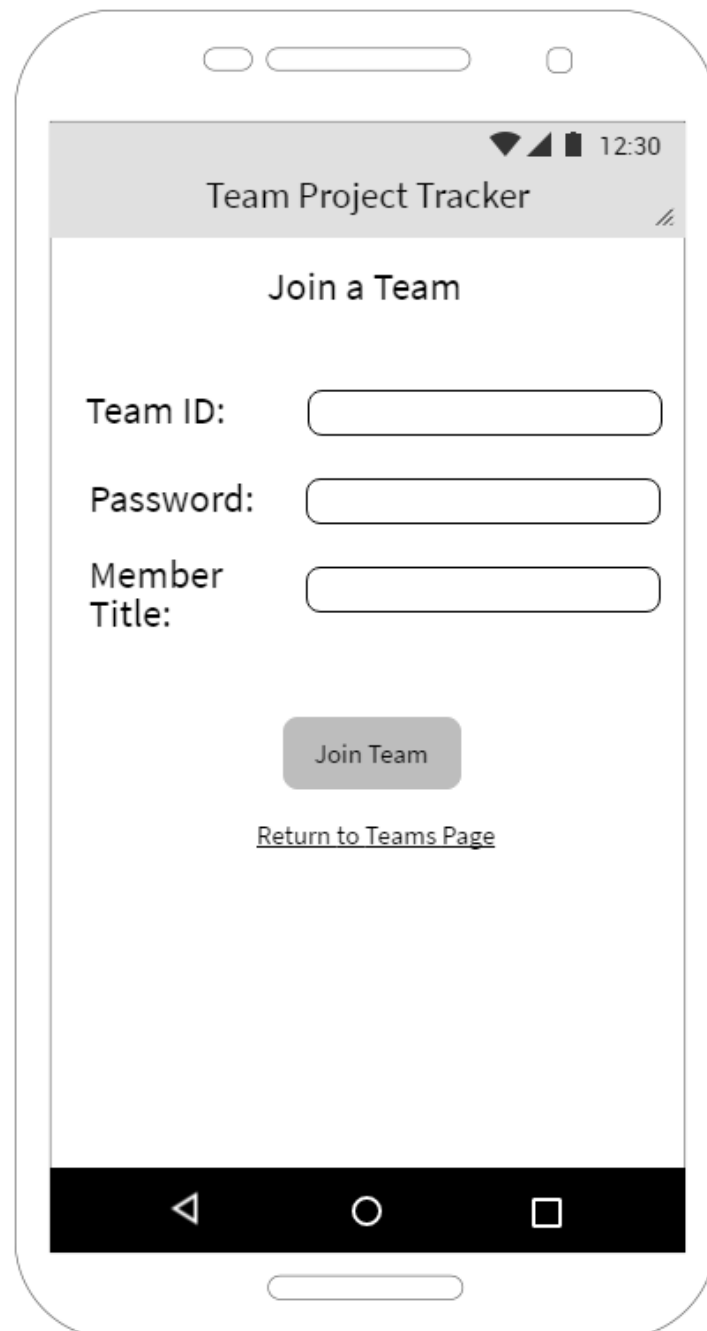


Fig 9. Successful Team Creation

4.1.3.6. Join a Team

To join a team, a member must enter the Team ID given to him by the administrator of the team along with the password of the team. There is another optional field where a user could enter a meaningful title for his role at the team. This can be later changed by the administrator.



The image shows a mobile application interface for 'Team Project Tracker'. The screen displays a 'Join a Team' form. At the top, the status bar shows the time as 12:30. The app title 'Team Project Tracker' is centered at the top of the screen. Below the title, the heading 'Join a Team' is centered. The form consists of three input fields: 'Team ID:', 'Password:', and 'Member Title:'. Each field is followed by a text input box. Below the input fields, there is a 'Join Team' button. At the bottom of the form, there is a link labeled 'Return to Teams Page'. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps buttons.

Fig 10. Join a Team

4.1.3.7. Team Dashboard – Member

This dashboard shows if you join a team as a member.

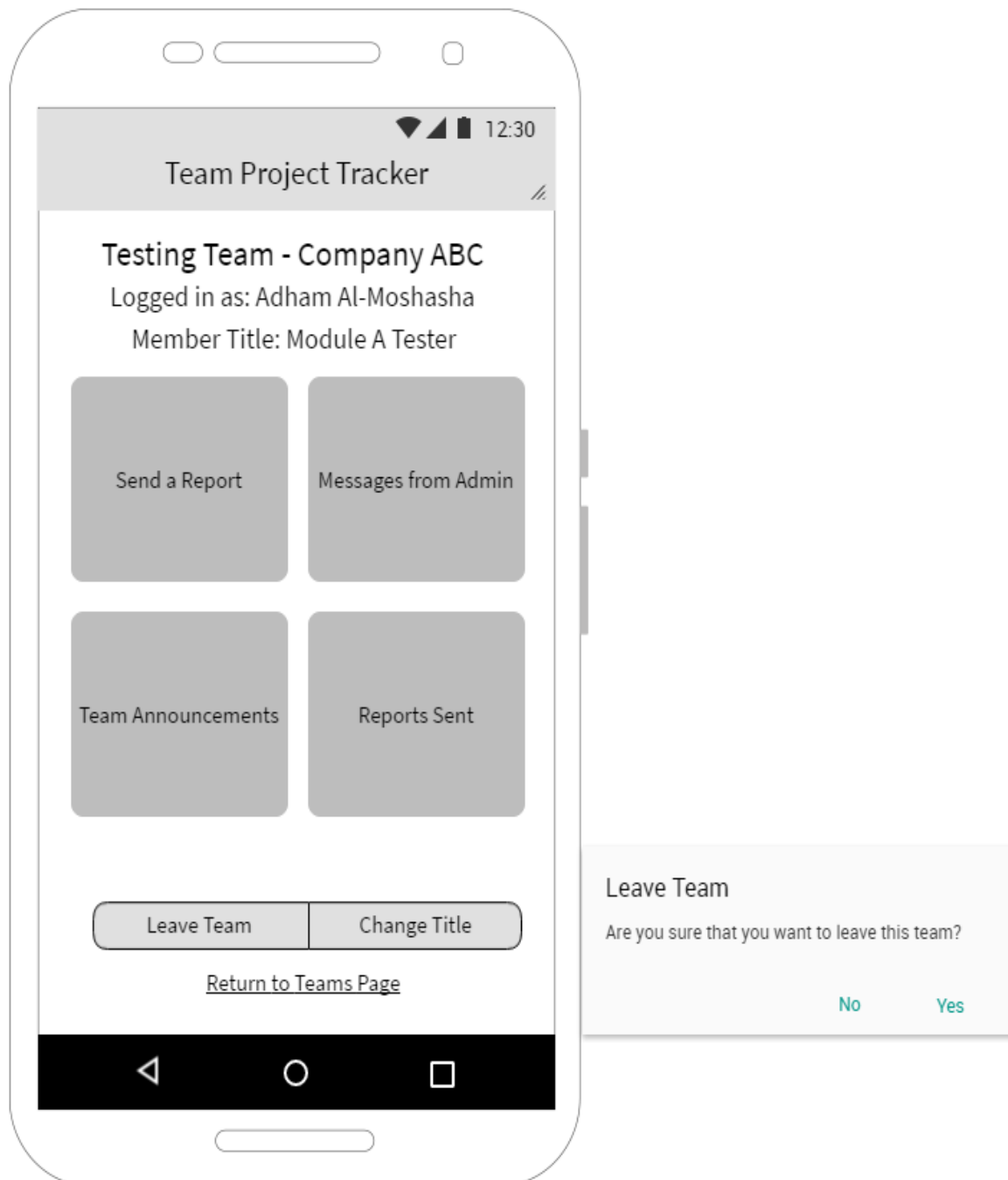


Fig 11. Team Dashboard – Member

4.1.3.8. Team Dashboard – Admin

This dashboard shows if you join a team as an admin.

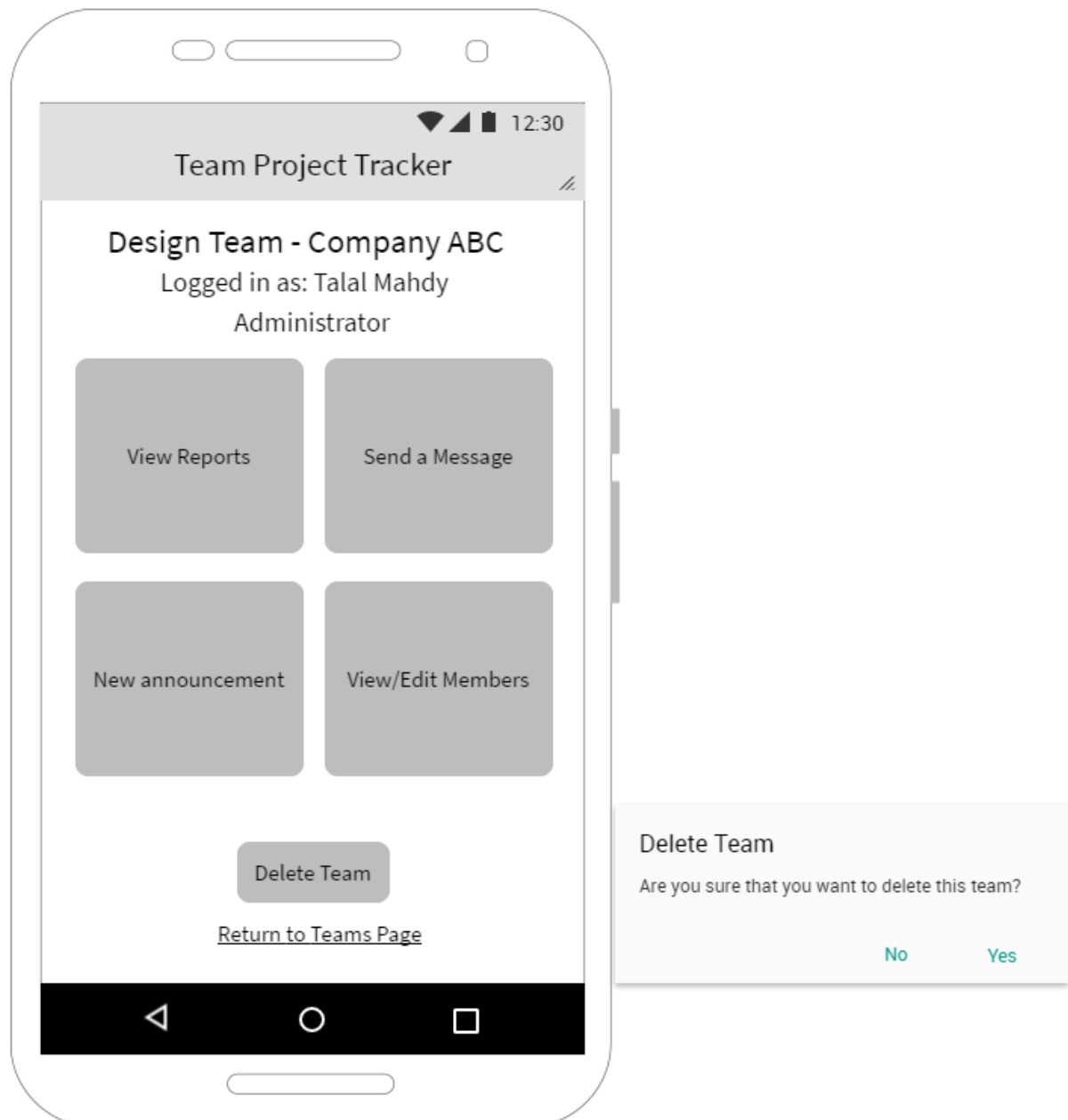


Fig 12. Team Dashboard – Admin

4.1.3.9. Send a Report

To send a report to the team leader, enter any number of hours under any number of days then choose a preexisting task or create a new one and then write a description of the task done.

The image shows a mobile application interface for "Team Project Tracker". The screen is titled "Send Report". At the top, there's a status bar with the time 12:30 and battery level. Below the title, there's a calendar for December 2017. The calendar has columns for days of the week (Mo, Tu, We, Th, Fr, Sa, Su) and rows for dates. The number 7 is highlighted in the first row, under the 'Su' column. A callout box points to this number, containing the text "7" and "total hours worked on day 7 shown". Below the calendar, there's a "Task Name:" label followed by a dropdown menu with the text "User selects a task or enters a new one". Below that is a "Description of Task:" label followed by a text input area. At the bottom, there are two buttons: "Reset" and "Send Report". Below these buttons is a link that says "Return to Teams Dashboard".

December 2017						
Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Task Name:

Description of Task:

[Return to Teams Dashboard](#)

7 total hours worked on day 7 shown

User enters hours worked on a task

Fig 13. Send a Report

4.1.3.10. Messages from Admin



Fig 14. Messages from Admin

4.1.3.11. Reports sent to Admin

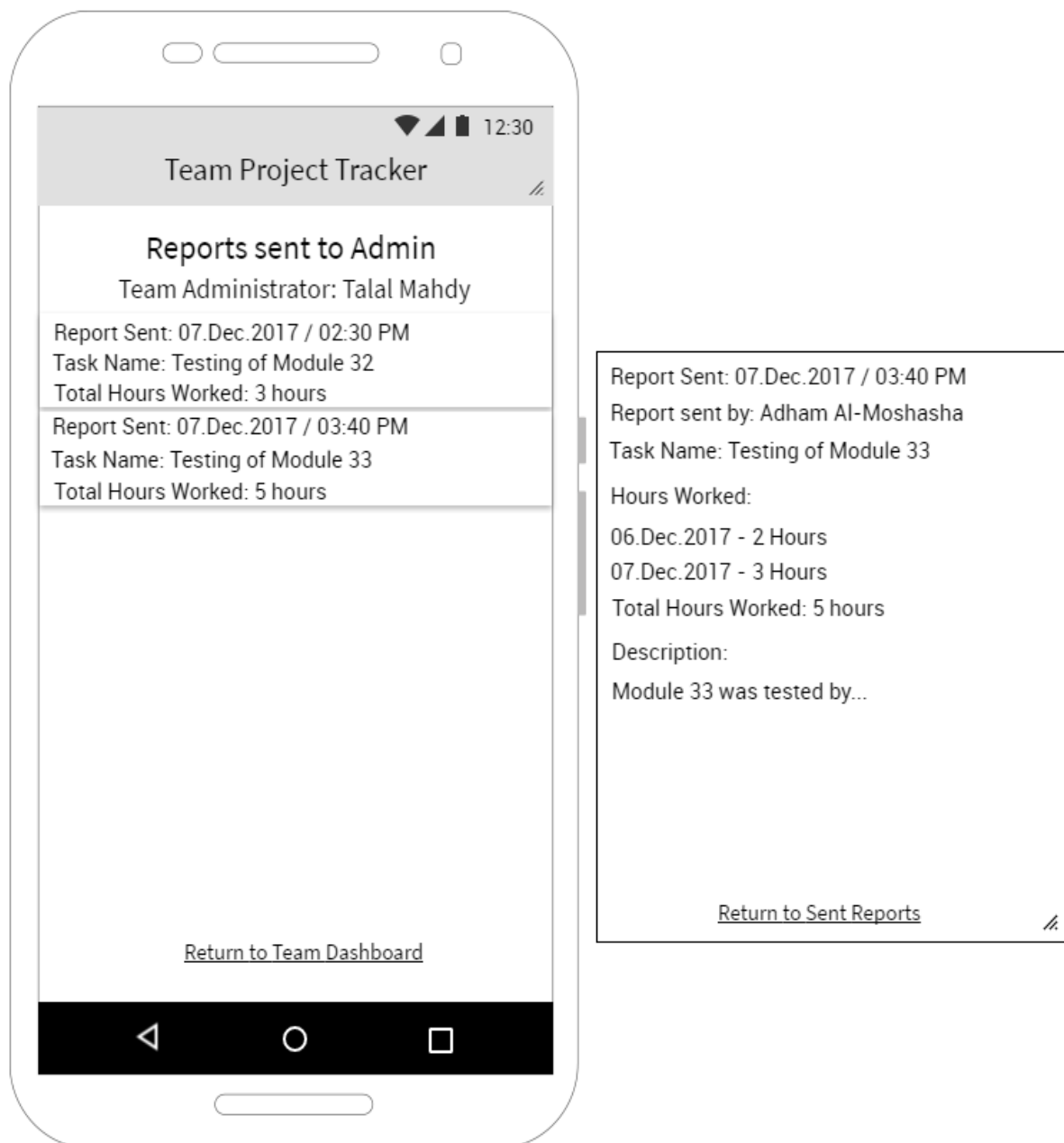


Fig 15. Reports sent to admin

4.1.3.12. Edit/View Users

On this page, an admin can remove a team member or change his title.

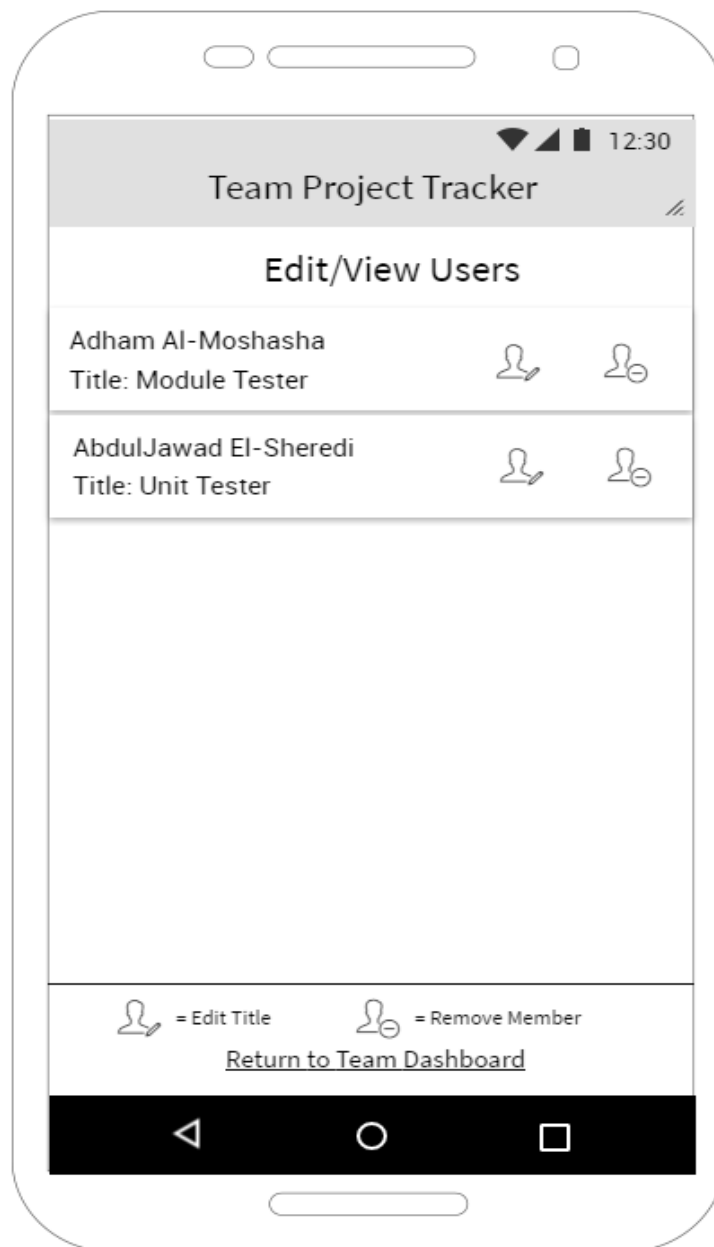


Fig 16. Messages from Admin

4.2. Low level design (components used)

4.2.1. Dataflow Diagram – Level 1

This dataflow diagram describes how data flows in the important report sending module of Team Project Tracker.

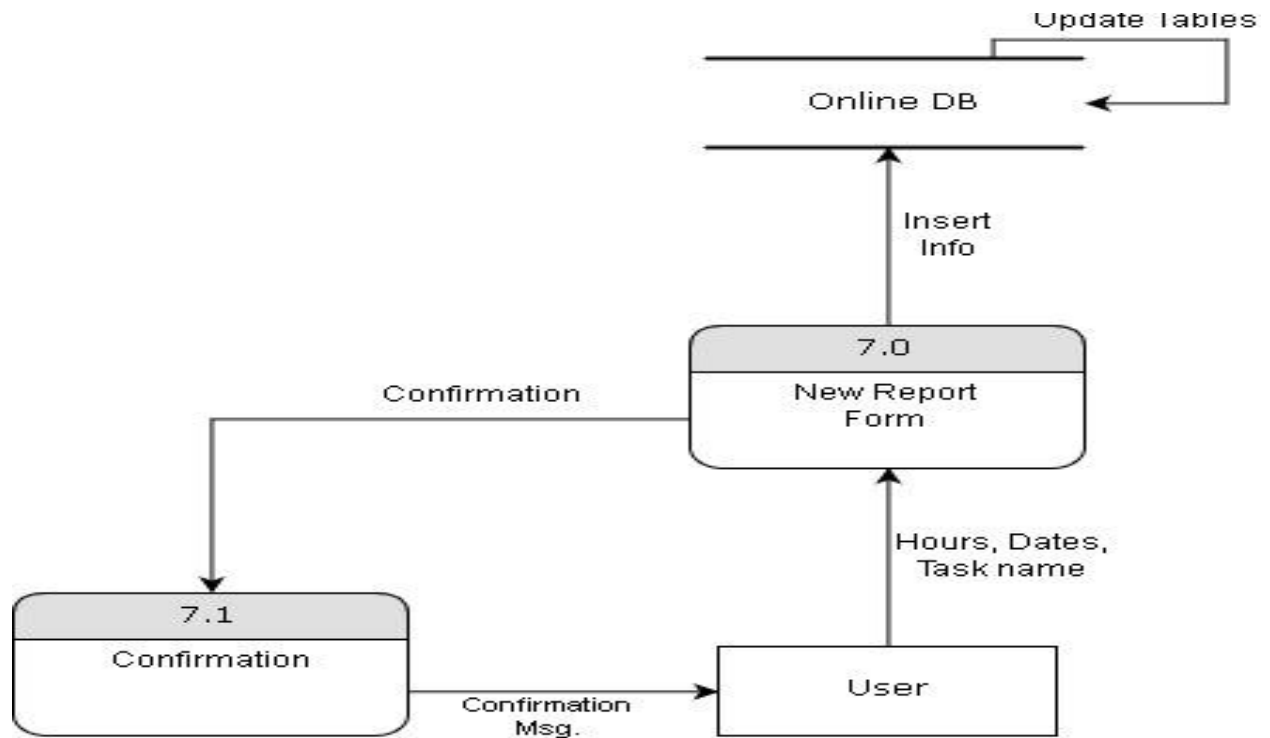


Fig 17. Level 1 Data Flow Diagram (Sending a Report)

4.2.2. Data Dependencies

This will be represented as an Entity Relationship Diagram in Figure 19. This diagram was created using the draw.io tool.

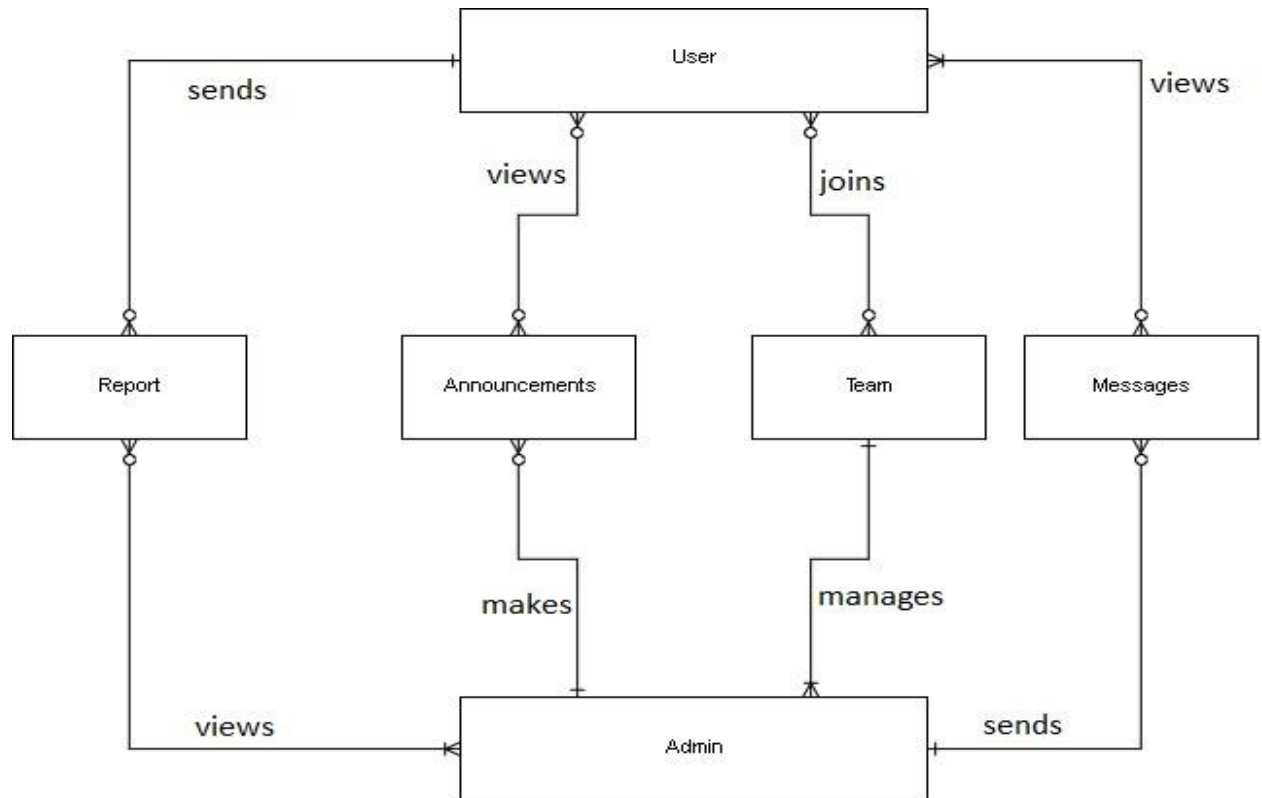


Fig 18. ERD for Team Project Tracker

4.2.3. Class Diagram

The class diagram in figure 20 shows the main classes of Team Project Tracker and the relations between them.

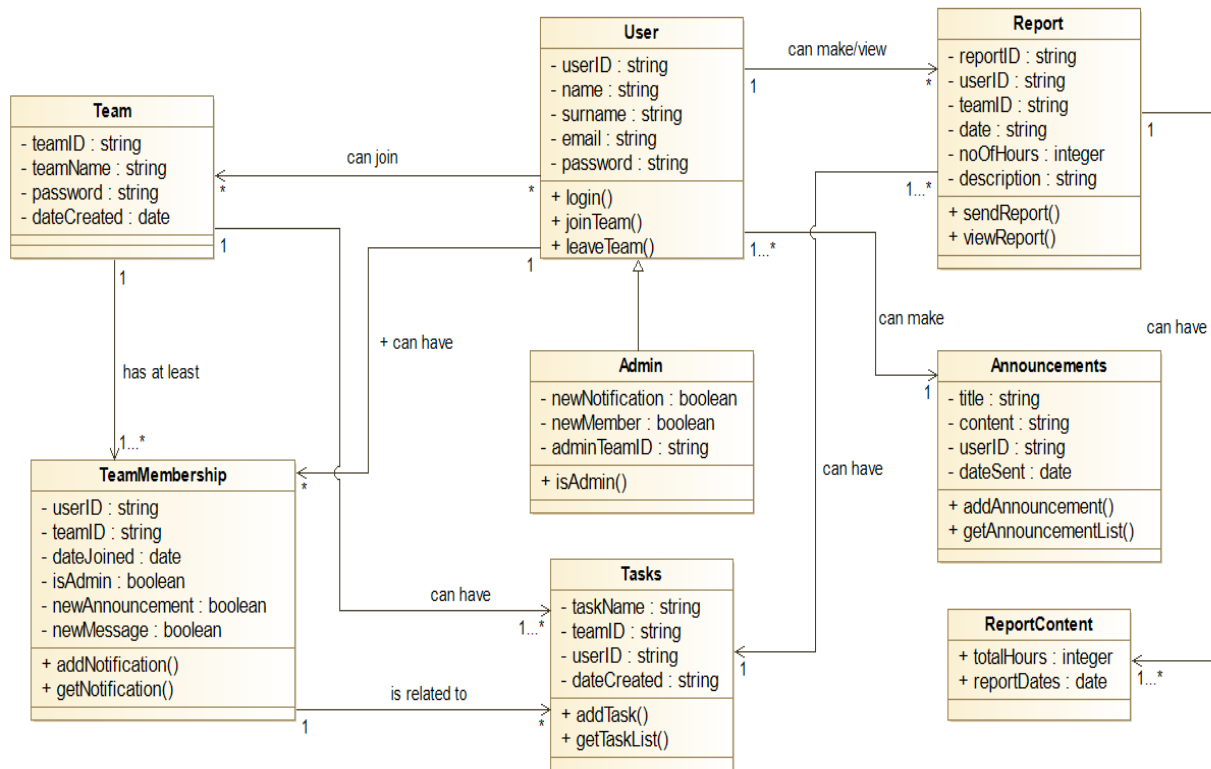


Fig 19. Class Diagram for Team Project Tracker

4.2.4. UML Interaction Diagrams

A use case diagram which was created using the Modelio tool is shown in figure 21. The two sequence diagrams shown in Figures 22 and 23 were created using the Modelio and draw.io tools respectively. An activity diagrams is shown in Figure 24 which was created using draw.io tool. A business process Model is shown in figure 25 which was created using the Modelio tool.

4.2.4.1. Use Case Diagram

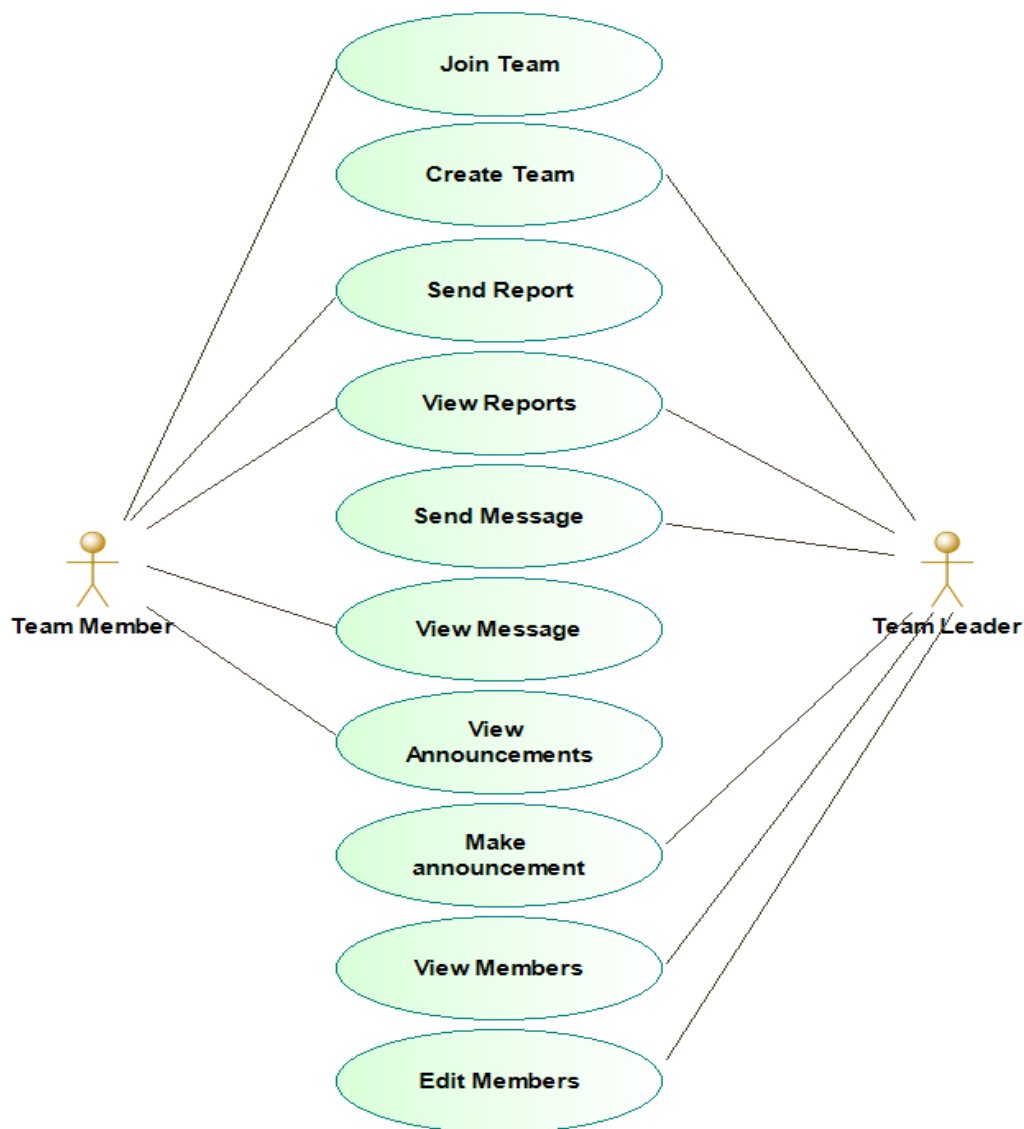


Fig 20. Use Case Diagram

4.2.4.2. Sequence Diagram – Login to System

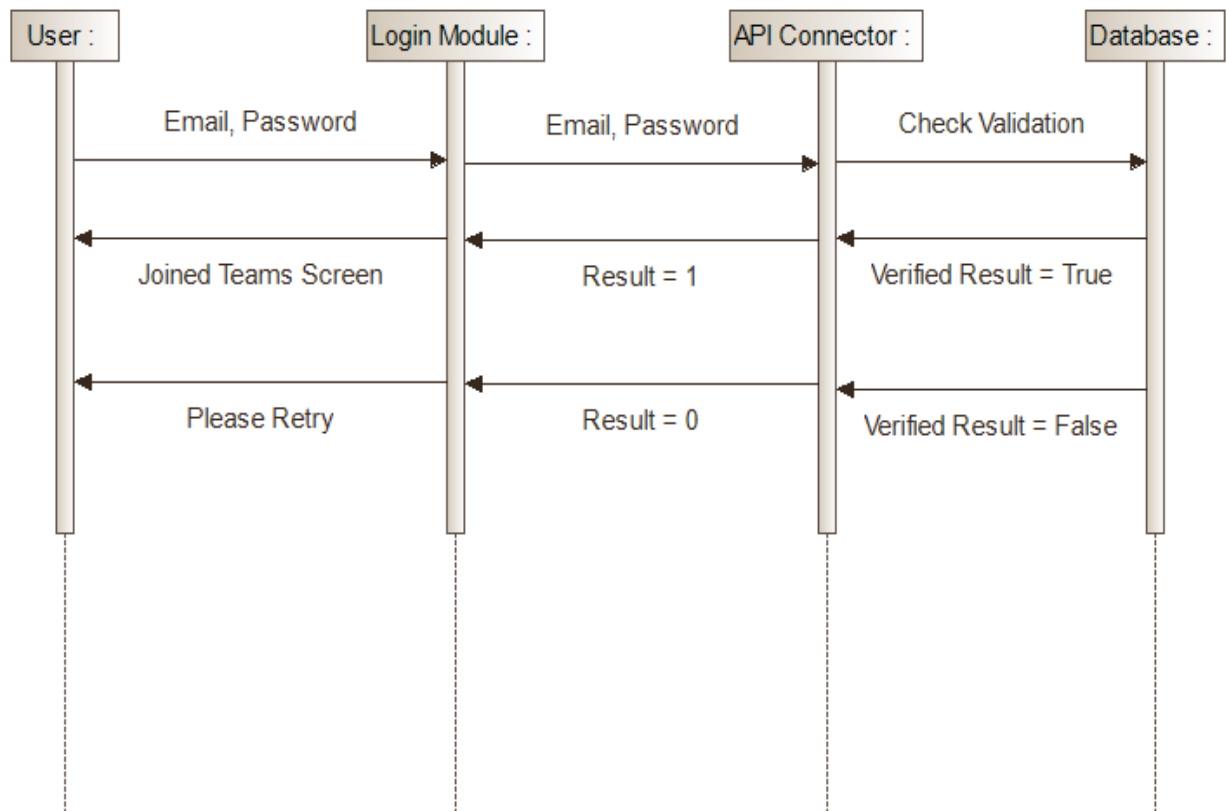


Fig 21. Sequence Diagram – Login to System

4.2.4.3. Sequence Diagram – Register a new User

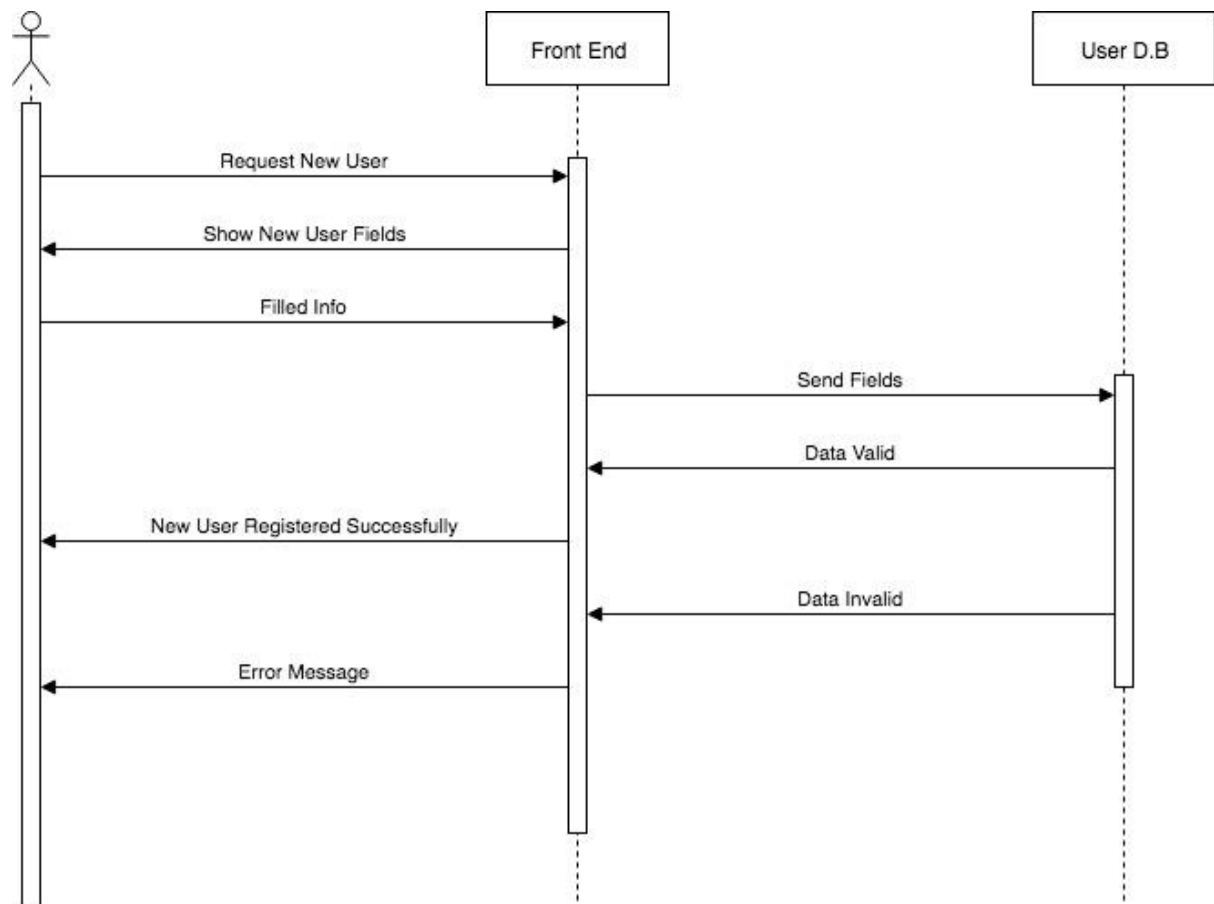


Fig 22. Sequence Diagram – Register a new User

4.2.4.4. Activity Diagram – Create a Team

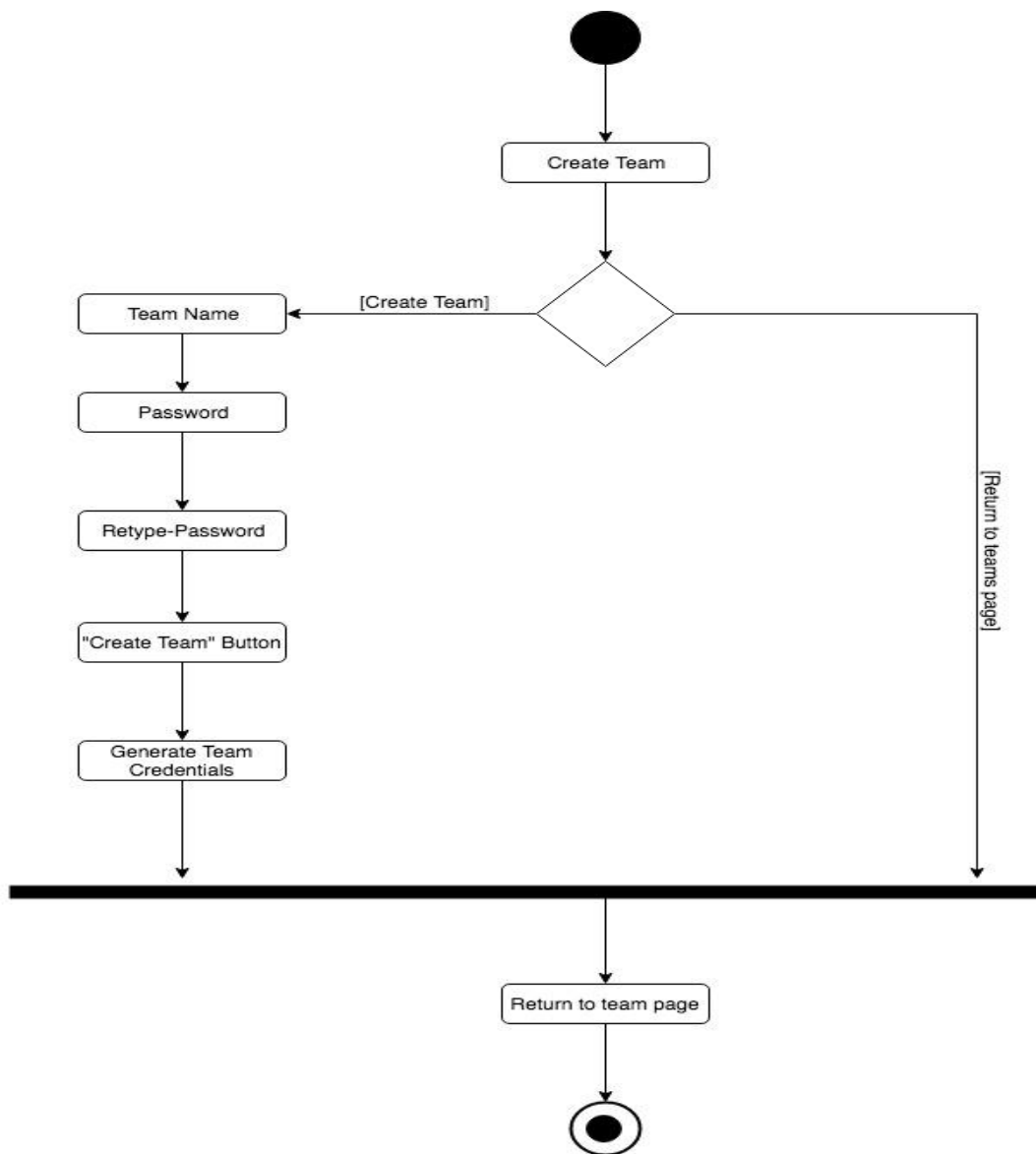


Fig 23. Activity Diagram – Create a Team

4.2.4.5. Business Process Model

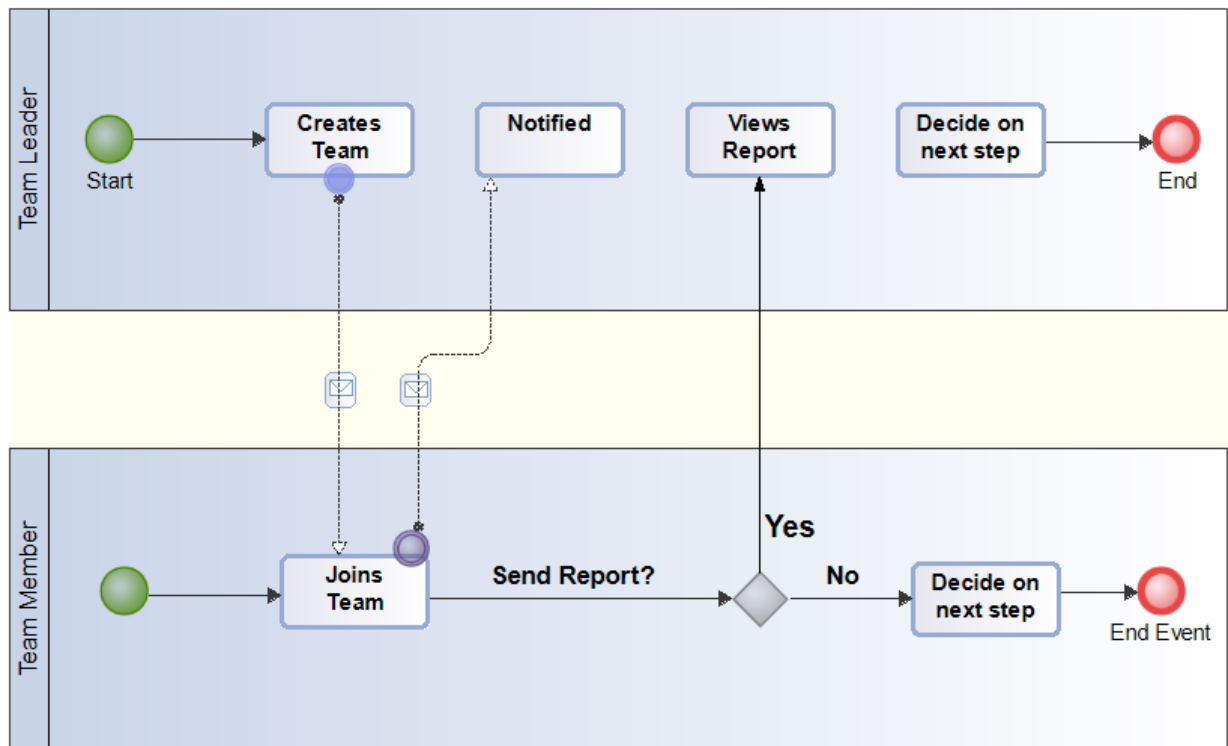


Fig 24. Business Process Model

5. IMPLEMENTATION

5.1. Tools, technologies and platforms used

C# Programming language, Microsoft SQL Server Management Studio, Visual Studio IDE, Microsoft Azure Cloud Services, ASP.NET RESTful Web Service, JSON

5.2. Use of Software Engineering Process Steps

These steps were carried out in order to build the project:

1. Determination of an ideal Software Development Approach: Incremental Model. This was discussed in Section 2.6 and is shown in figure 25.

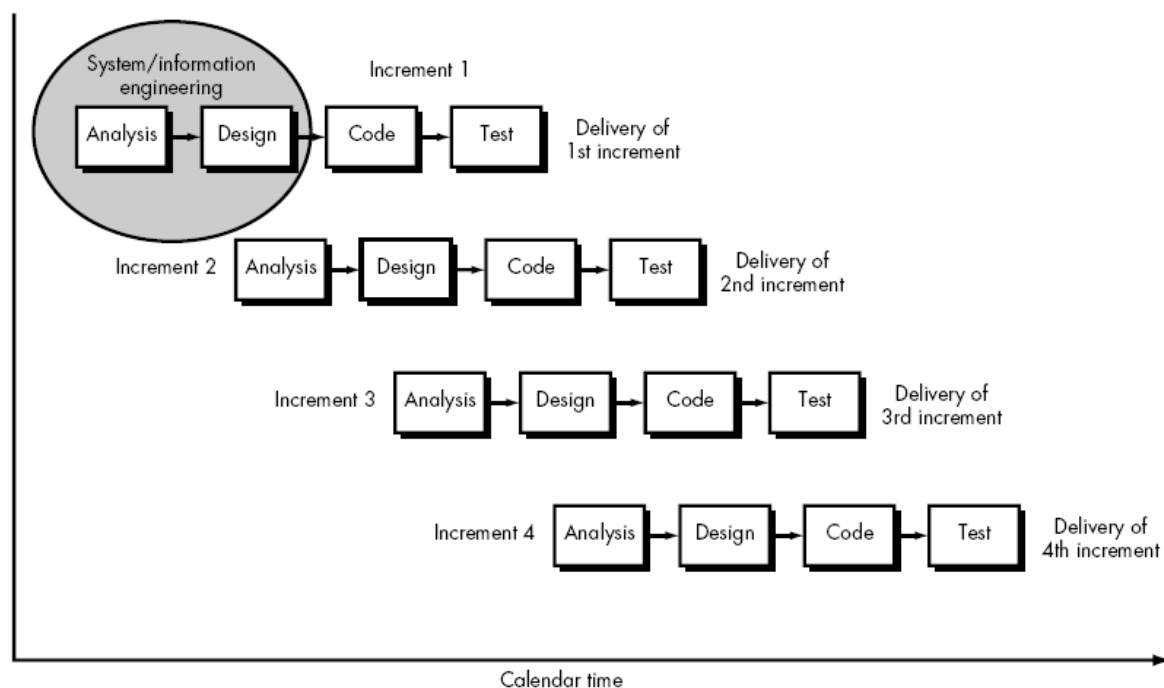


Fig 25. Incremental Method of Development

2. Make a Software Development Plan with the help of Microsoft Project tool.

3. Arrange the features of the system according to their importance.

High Priority:

- a. Registration
- b. Login
- c. Send Report
- d. Edit/View Members

Low Priority:

e. Team announcements

f. Send a message

4. Starting with the most important modules, write their specific requirements and prepare the SRS.

5. At the same time, prepare the architecture of the system and overall design and write the SDS. Also, prepare the designs for the important modules to be handed over to the programmers using the Requirements made by the Requirements Engineers.

6. When the requirements and design of the important modules are over, start the development of the most important modules.

7. Begin the development of the next modules by the time the testing of the already developed modules is over.

8. The steps above are repeated until development of all modules is complete and the system is integrated.

9. The system as a whole is verified and validated and then deployed.

10. Updates and Maintenance to the system is periodically done.

5.3. Algorithms

Algorithm 1: Login

Begin

Enter password

if password correct then log user in

else output “wrong password”

End

Algorithm 2: Send Report

Begin

if user chooses (any number of dates written with hours && writes description)

then send report to admin

else do not send report to admin

End

Algorithm 3: Join Team

Begin

user chooses to join a team

system asks user for teamID, password and title

if (teamID or password) = correct

then join team

else cannot join team

End

Algorithm 4: Leave Team

Begin

user chooses team dashboard

if user chooses leave team

then leave team

else do nothing

End

Algorithm 5: Delete Team

Begin

admin chooses team dashboard

if admin chooses delete team

then delete team

else do nothing

End

5.4. Standards

Project Proposal Form: TUBITAK

Software Requirements Specification Document: IEEE 830-1998

Software Design Specification Document: IEEE 1016-1998

Security: SHA1 Hashing

Implementation: Google Development Guidelines¹²

5.5. Detailed description of the implementation

As the design and requirements step for each module is complete, the implementation step for that module begins. The implementation of the database is also considered in this Section. The application was developed using mainly using the C# Programming Language. There are a total of 18 pages in the implementation of the code on Visual Studio. There are also a lot of layout XAML page files on Visual Studio which work alongside the backend C# code.

There is only one online database hosted online by Microsoft Azure Cloud Services Platform which is managed by a database administrator using Microsoft SQL Server Management Studio tool. When the user runs the application, the first page shown is the login page and the application makes sure that a connection between the application and the Web Service is established. The Web Service was created using ASP.NET RESTful Web API and is also hosted online on Microsoft Azure. After a user clicks the login button, the application uses the API class responsible for communication with the Web Service and sends the login credentials to the web service. Then the web service receives the information and forwards it to the Database to check if the database contains any user with the same email and password and then provides the application with a result, either True (1) or False (0). If the result is true, the web service saves the userID on the session storage which will be used to process any future requests until the user chooses to sign out or the session key expires (after 15 minutes of inactiveness). The web service transfers all essential data and communication with the application through the JSON standard which the application reads as a string and de-serializes it to classes using the newtonsoft JSON library. Also, the user passwords are encrypted using the SHA-1 Standard on the Online Database.

The purpose of the online database is to update all the application related data that user interacts with. The online database is controlled by a trustworthy database administrator. Only the Web Service can modify the Database and the modification can only be done after processing the user's requests and checking the userID and info. An internet connection is required throughout the runtime of the application since each request by the user checks if the session key is valid and also checks the session storage. The users can only access the info that is related to them.

When the user requests any data related to them, the Web Service makes sure to check if the user is really a member of that team and if the request needs administrator privileges, the web service makes sure that the user has these privileges. All the processing is done by the web

service and the main job of the application is just to show the requests and receive information from the user.

The implementations in Visual Studio along with the databases are shown in figures 26 to 34.

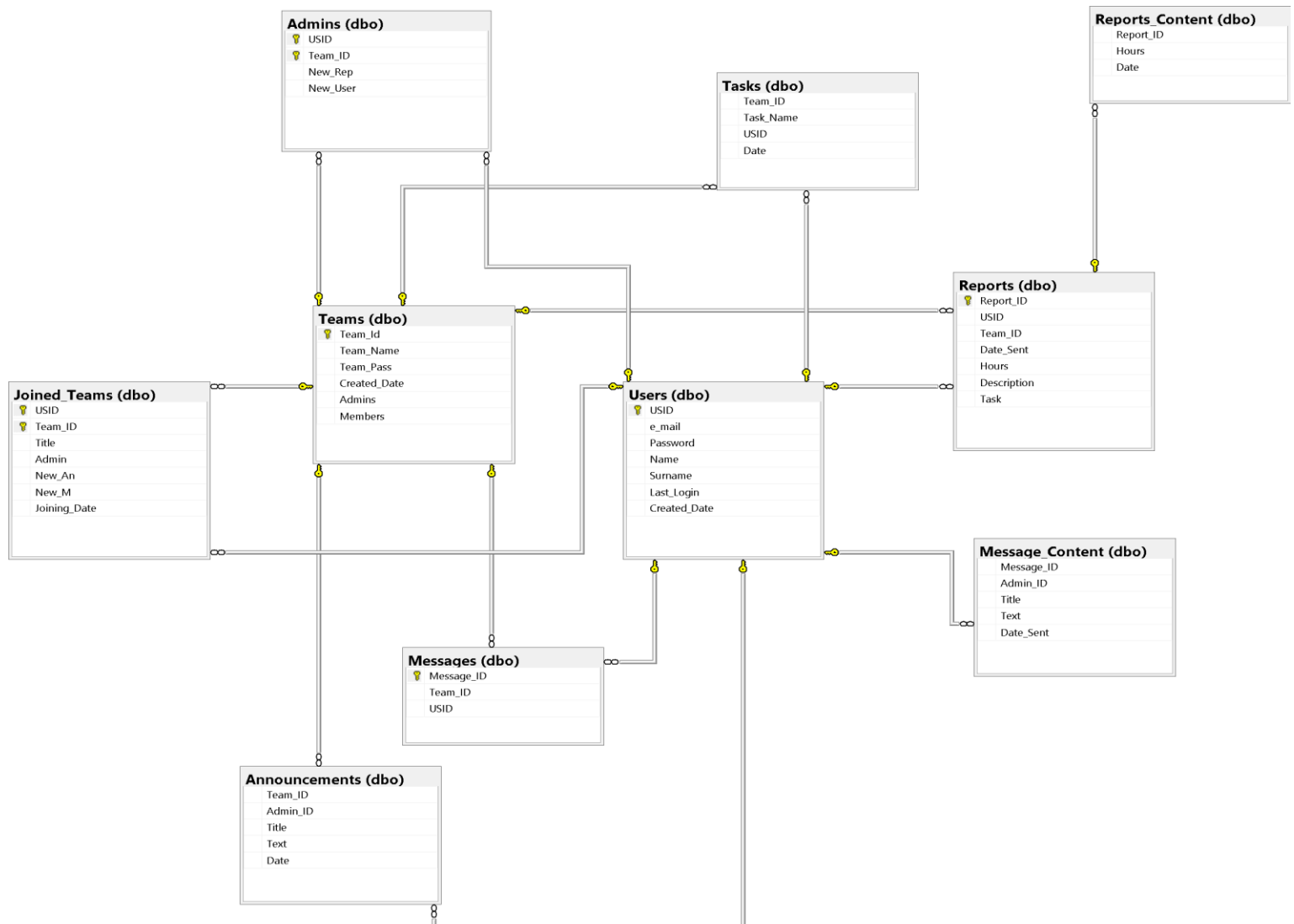


Fig 26 (i) . Database Implementation in Microsoft SQL Server Management Studio

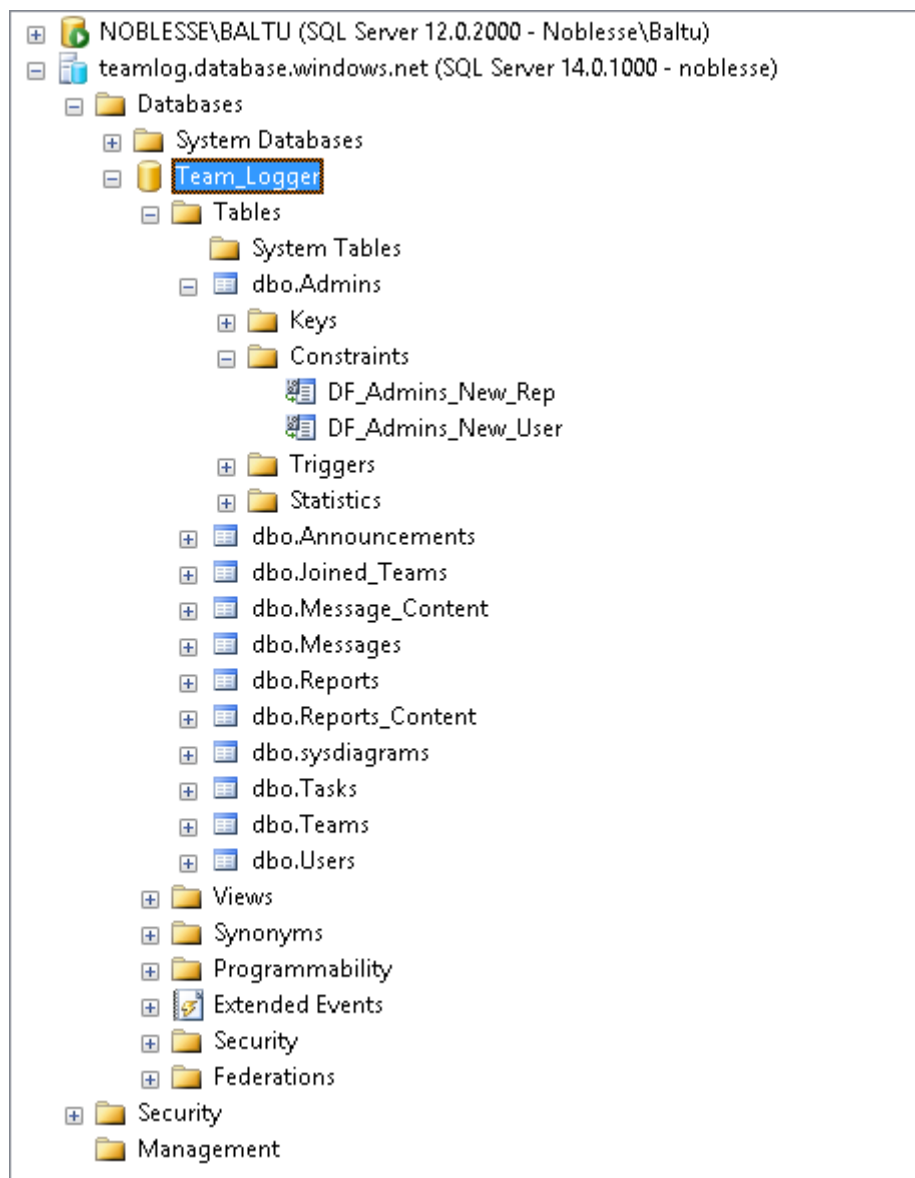


Fig 26 (ii) . Database Implementation in Microsoft SQL Server Management Studio

Microsoft Azure SQL databases > Team_Logger - Data explorer (preview)

libyan.noblesse@gmail... LIBYANNOBLESSEMAIL (DEF...)

Team_Logger - Data explorer (preview)

+ New

Dashboard

All resources

Resource groups

App Services

SQL databases

SQL data warehouses

Azure Cosmos DB

Virtual machines

Load balancers

Storage accounts

Virtual networks

Azure Active Directory

More services

Login Edit Data (Preview) + New Query Open query Save query Feedback

Showing limited object explorer here. For full capability please open SSDT.

Tables

dbo. Announcements

Team_ID (nvarchar, not null)

Admin_ID (nvarchar, not null)

Title (nvarchar, not null)

Text (ntext, not null)

Date (datetime2, not null)

dbo. sysdiagrams

dbo. Users

dbo. Teams

dbo. Tasks

dbo. Reports

dbo. Reports_Content

Report_ID (nvarchar, not null)

Hours (int, not null)

Date (date, not null)

dbo. Messages

dbo. Messages_Content

Query 1 X

Run Cancel query

1 select * from Teams

TEAM_ID	TEAM_NAME	TEAM_PASS	CREATED_DATE	ADMINS	MEMBERS
0304e	g	gg	2017-12-24T0...	0	1
040cf	mokbel	123	2017-12-26T1...	1	1
08952	h	h	2017-12-24T2...	1	1
08c65	baltu team	z	2017-12-22T1...	1	3
0bbd2	g	g	2017-12-24T0...	0	1
11b10	j	y	2017-12-24T1...	0	1

Query succeeded | 1s

Fig 27. Database Implementation on Microsoft Azure Cloud Services

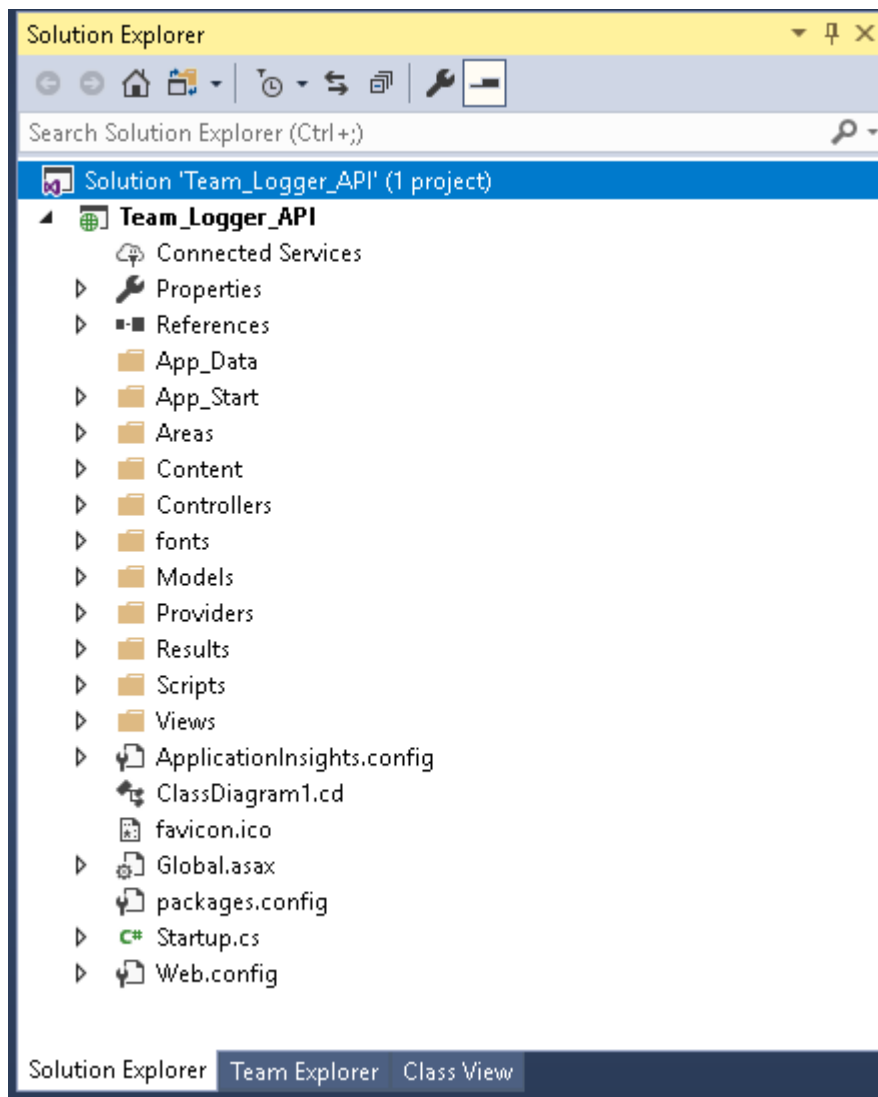


Fig 28. Visual Studio Project Files

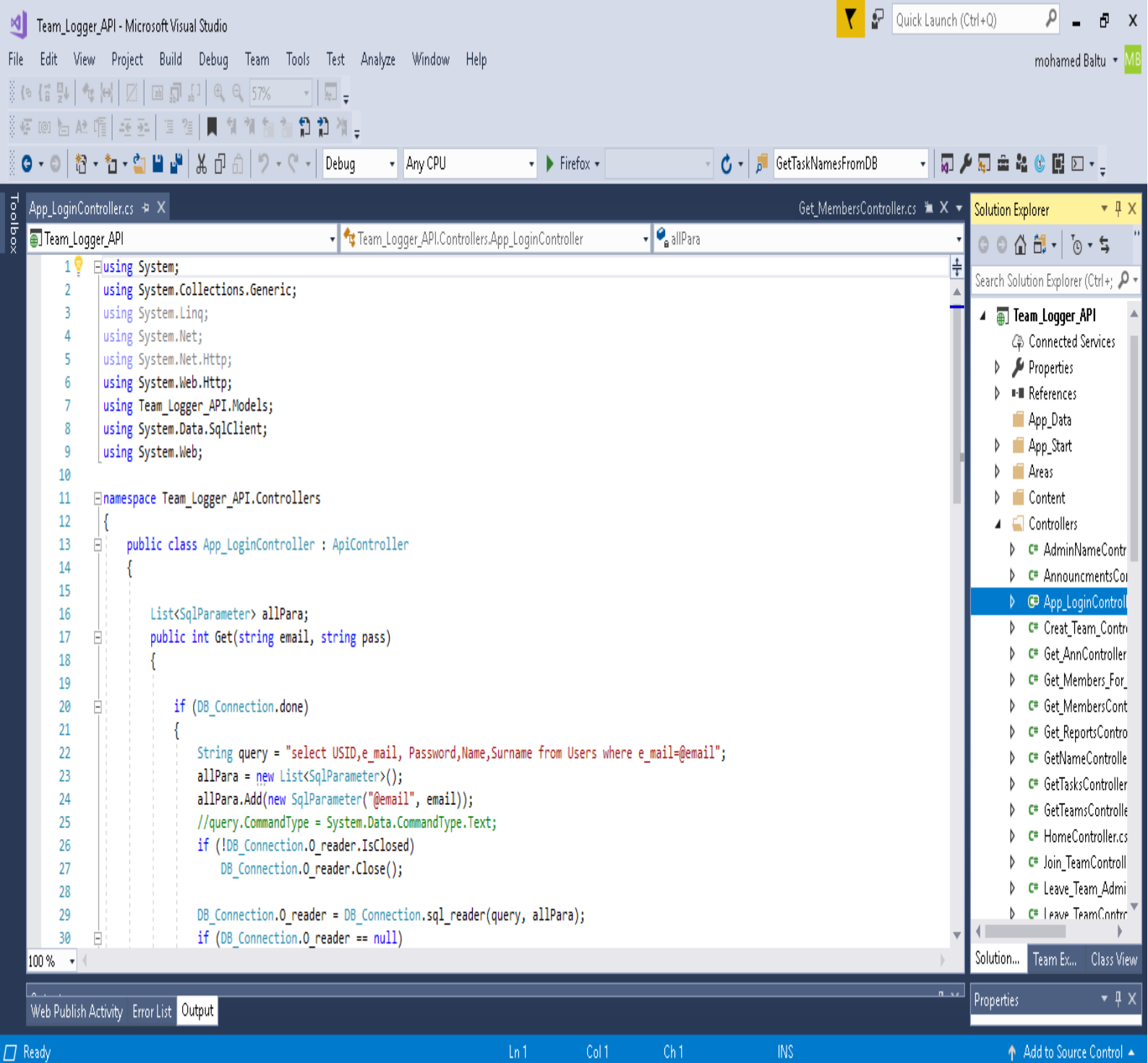


Fig 29 (i). Implementation of ASP.NET RESTful Web Service

ASP.NET Web API Help Page x Data explorer (preview) - Micro x +

teamloggerapi20171216093012.azurewebsites.net/Help

Project Tracker API Home API

ASP.NET Web API Help Page

Introduction

Provide a general description of your APIs here.

Get_Ann

API	Description
GET api/Get_Ann?Team_ID={Team_ID}	No documentation available.

Announcements

API	Description
GET api/Announcements?Title={Title}&Team_ID={Team_ID}&Text={Text}&Date={Date}	No documentation available.

GetTasks

API	Description
GET api/GetTasks?Team_ID={Team_ID}	No documentation available.

New_Account

Fig 29 (ii). Implementation of ASP.NET RESTful Web Service

6. TESTING

Modules of the application were tested manually for any bugs, crashes as well as any unexpected output results. Expected outputs were compared with the outputs perceived during the testing and subsequent changes and modification to the code were made in case of mismatch. The validity of the end product has been verified by installing the software on several android devices with different specifications in order to assess visuals, resolution, colors and compatibility of the application. This method also allowed the use of these android devices to test the app during its development, not only to validate the end product. Testing results showed an overall good standing of the end product with major features of the application working smoothly without crashes or unexpected outputs. In addition, Android 4.3 and older versions showed no compatibility with the application. However, some minor bugs might be present that require further testing after the end product release due to the large number of variables that might affect the application's various features and functions.

Table 14. Test Case 1

Test Case ID	TC-01
Test Case Name	Login test
Pass/Fail Criteria	Pass: user enters correct email, password -> login user enters wrong credentials -> "retry"
Input Data	Numeric/Alphabetic/Symbolic Password
Test Procedure	Expected Output:
Step 1: Enter correct email,password	Logs in to system
Step 2: Enter wrong email/password	Displays a message "wrong login information"
Comments	Test passes successfully

Table 15. Test Case 2

Test Case ID	TC-02
Test Case Name	Send Report Test
Pass/Fail Criteria	Pass: a report successfully sends from a user account to a team leader account Fail: report fails to send
Input Data	Report inputs (dates + hours + Text)
Test Procedure	Expected Output:
Step 1: Login to team member account. Create a report containing “05.12.17: 5 hours” “31.12.17: 8 hours”, write a task & description and send report.	Report successfully sent.
Step 2: Login to the team leader account. Click on View Reports. Click on the report.	Displays all sent reports, displays clicked report in specified format.
Comments	Test passes successfully

Table 16. Test Case 3

Test Case ID	TC-03
Test Case Name	Edit Member title test
Pass/Fail Criteria	Pass: Member Title successfully changes and displays to user Fail: Member title does not change or display
Input Data	New Member title
Test Procedure	Expected Output:
Step 1: login to team leader account	Successfully logs in
Step 2: click on a team and select View/Edit Members	View/Edit Members menu opens
Step 3: click on a member and change his title	Member title successfully changes
Comments	Test passes successfully

7. USER GUIDE OF THE SYSTEM

When the user launches the application, the login page shown in figure 30 will appear providing input fields for Email and Password, as well as two buttons for logging in and for new registration. If the user inputs incorrect login credentials, the application will provide an error. If the user wishes to register and clicks on “REGISTRATION” button, the application will navigate to figures 31, where the user is prompted to fill in the required fields. The user may also go back to the previous page by clicking the “GO BACK” button. After the user successfully logs in, the application will navigate to the Teams Dashboard page shown in figure 32, where the user can view and select joined teams, create team, join team and logout, Administrator icon will show next to teams where the user is set to be the admin, a notification icon will show next to teams that has new unread notifications. The user may create a new team by clicking on Create team button which will navigate to Team Creation page shown in figure 33, after the team is created successfully, figure 34 will be shown with auto generated team id and password, which will be used by registered users wishing to join the created team. Figure 35 shows the page where users are required to enter the given auto generated credentials. Figures 36&40 shows the main dashboard for both team members and admins respectively where they can click on multiple buttons, each one having a different functionality as shown in the figures below.

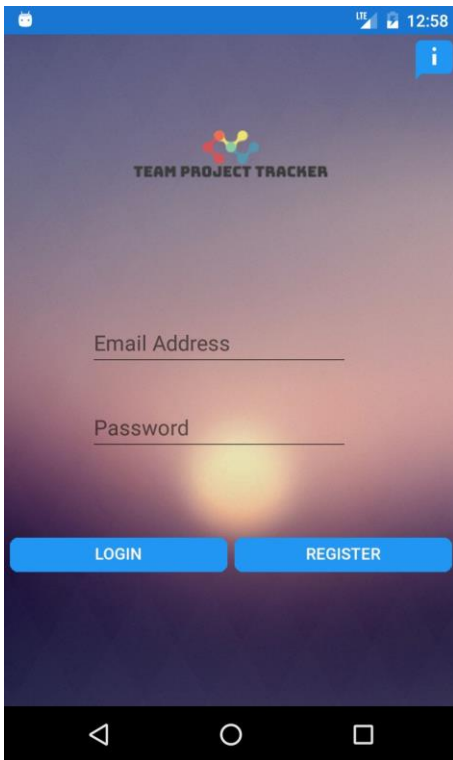


Fig 30. Login/Reg_Guide

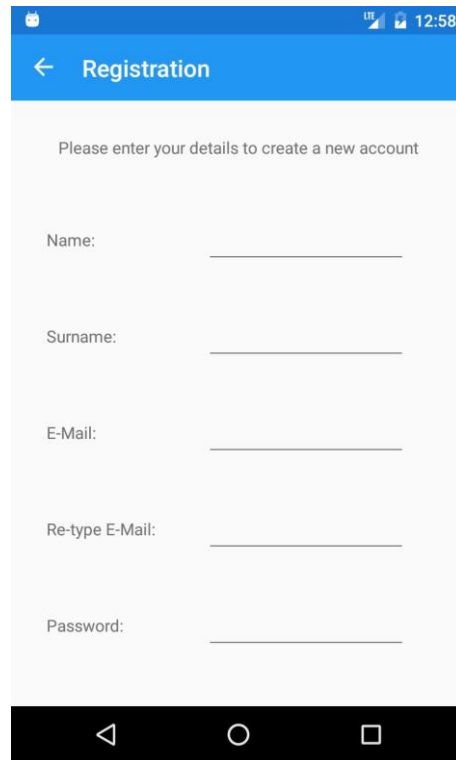


Fig 31. Registration Page_Guide

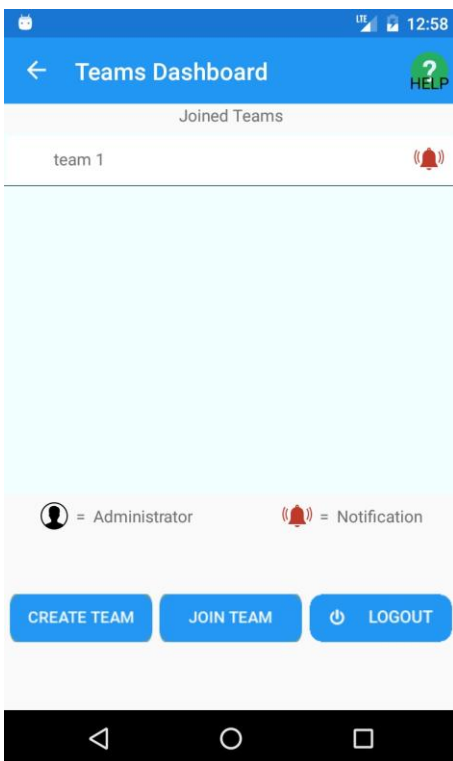


Fig. 32 Teams Dashboard_Guide

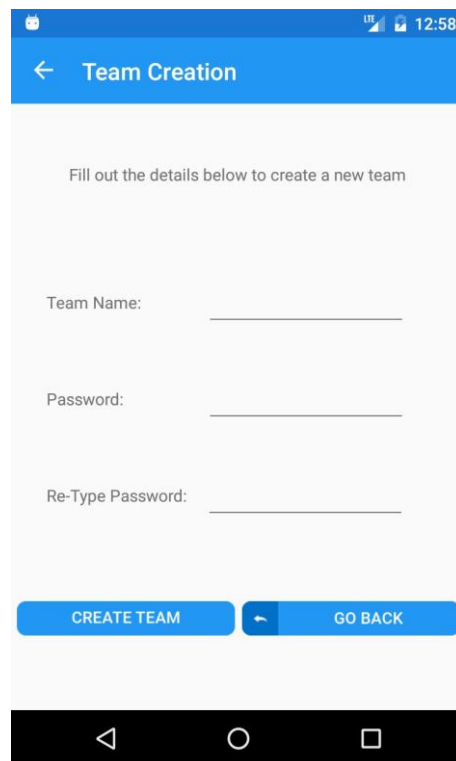


Fig. 33 Team Creation_Guide

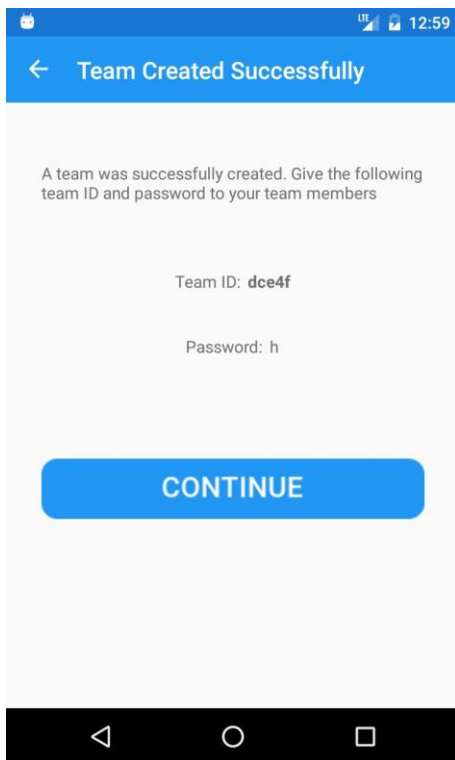


Fig 34. Teams credential_Guide

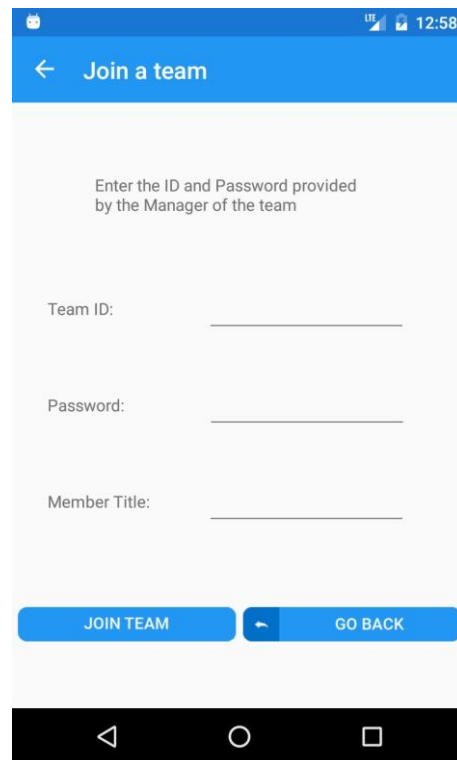


Fig 35. Team Join_Guide

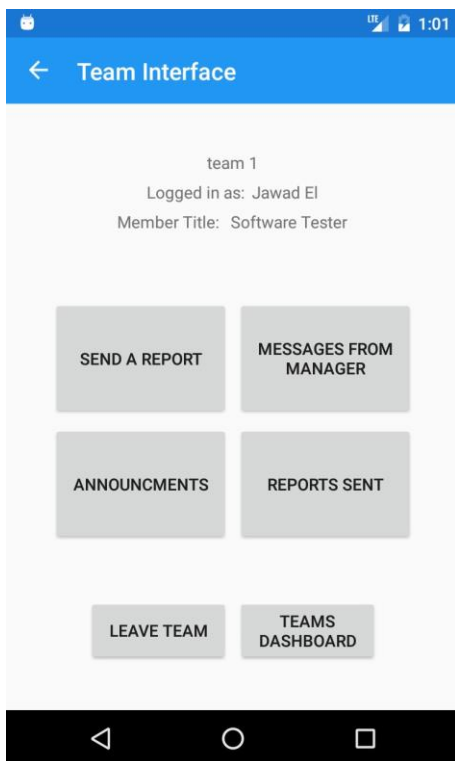


Fig 36. Member dashboard_Guide

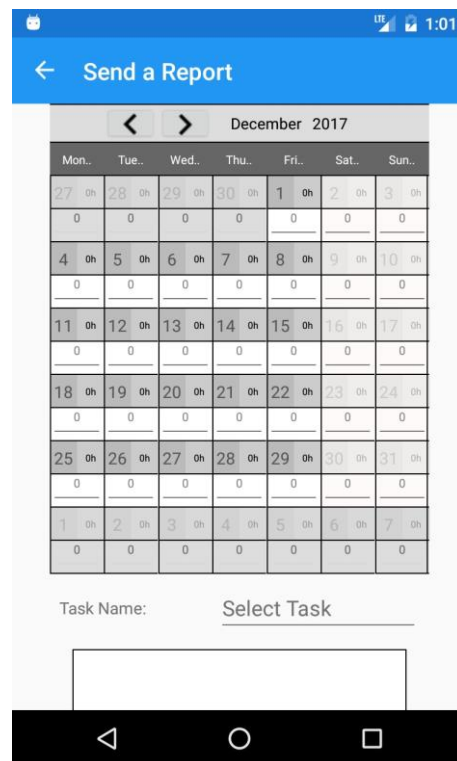


Fig 37 (i) Send a Report_Guide

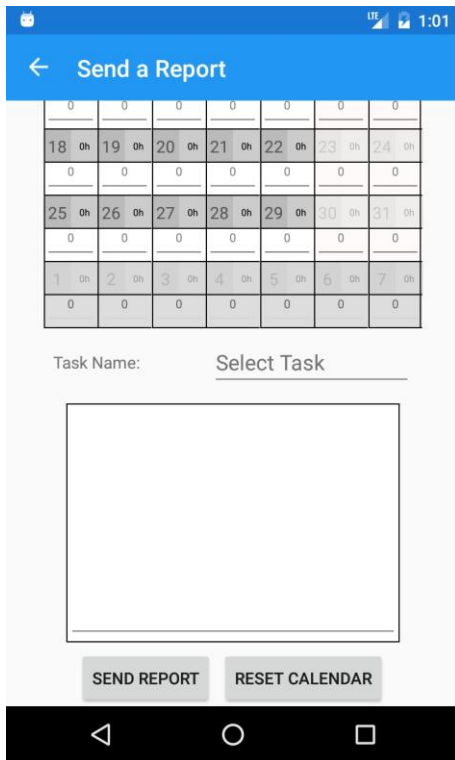


Fig 37 (ii) Send a Report_Guide

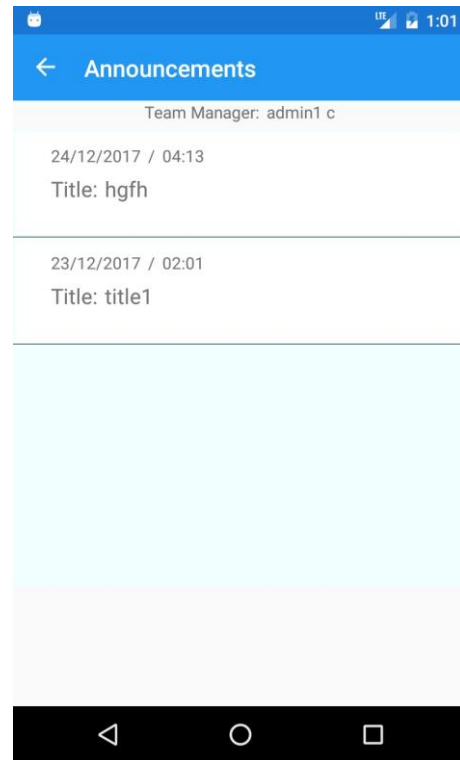


Fig 38. Announcements_Guide



Fig 39. Reports Sent_Guide

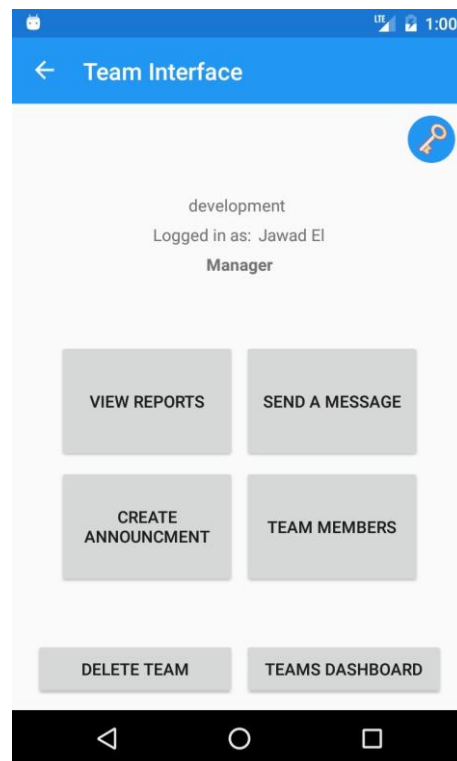


Fig 40. Admin dashboard_Guide

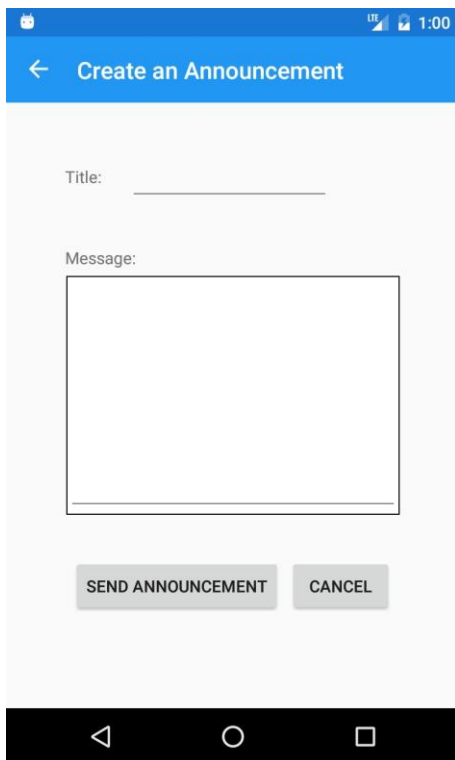


Fig 41. Create Announcement_Guide

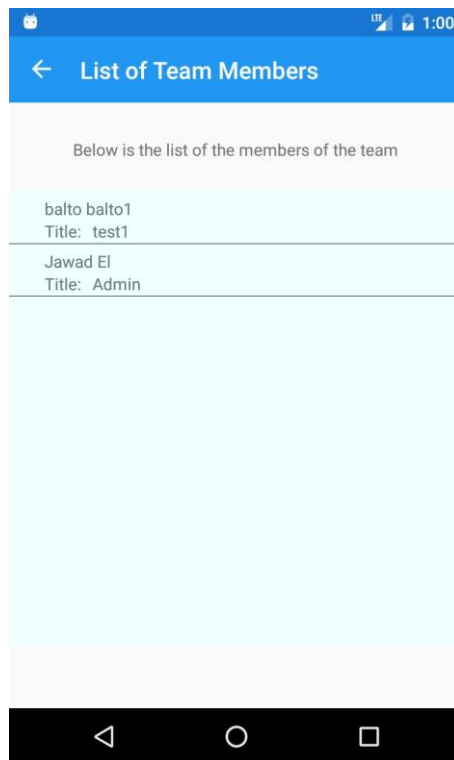


Fig 42. List Members_Guide

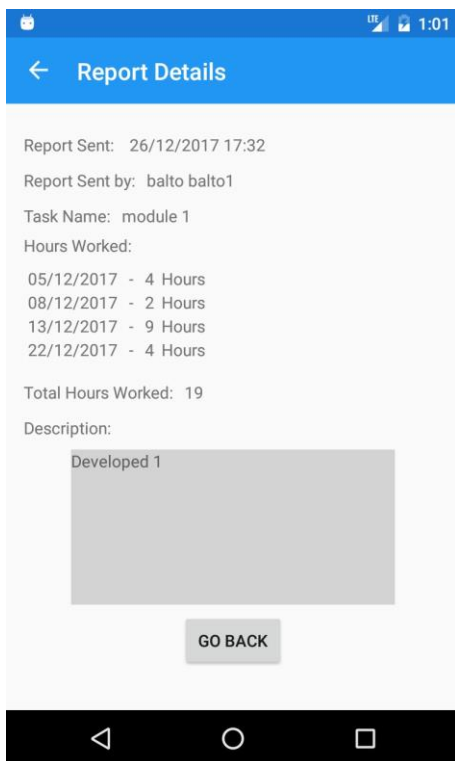


Fig 43 (i). Report Details_Guide

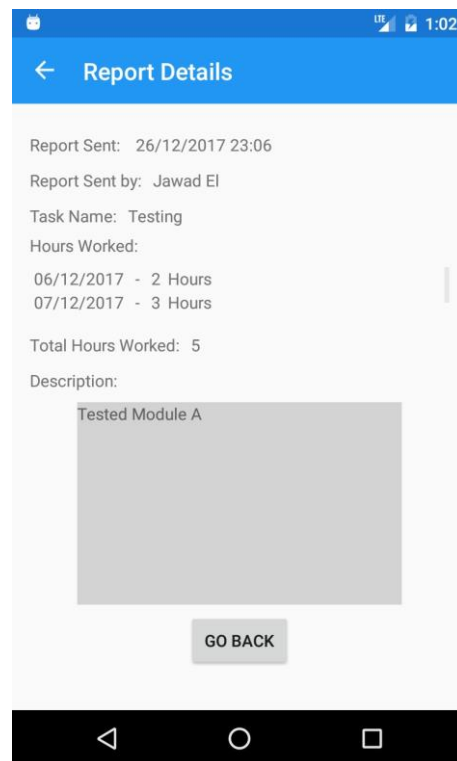


Fig 43 (ii). Report Details_Guide

8. DISCUSSION

The Team Project Tracker can be of great relief to Team Leaders and Team Members around the World. Team Project Tracker is a not-for-profit and an open source project which has a good probability for increasing the performance of teams by assisting them in increasing their performance in various fields such as time management, organization and productivity. The report sending module of the application is the most important section of Team Project Tracker. It allows for an organized form of communication where a team member can send any amount of reports in a static format. These reports contain very useful information for managers or team leaders to review. Information like the hours worked on a specific day, the task that has been worked on, the description of the task is crucial in an everyday working environment. It is not only crucial for a working environment, it is also crucial for any type of team such as a football team. A coach along with his team may use this application to organize training sessions for example. This is so useful especially since the timings of the training sessions can change or can differ from player to player. The application also allows team leaders to send specialized messages to the team members. Team members can also view important announcements made by the team leader. When creating a team, the application randomly generates a teamID for more security and due to the fact that many teams with the same name can be created. A feature unique to the team leaders is the View or Edit Members where a team leader can remove a member or change his title. Therefore, Team Project Tracker can be of great benefit to the employees around the world. It acts as an employee package and provides much functionality that a team leader and member need to succeed.

9. CONCLUSION

The Team Project Tracker is an Android mobile application made for employees or company teams around the world. The application can act as a complete package for an employee and can be really useful in keeping the team members and leaders productive and organized. It is a great application considering it has been developed in only 4 months which is a relatively short time. The Team Project Tracker has the potential to grow further and further as new features gets added in the future and I predict a bright successful future for the application. Team Project Tracker can also act as an example to other organizations on how a properly designed system acts like and other organizations could benefit from the application's designs and functionalities in producing their own applications. Software Engineering students may also learn a lot of things by referring to this application and the development of this application helped me personally by giving me the chance to learn the C# programming language, XAML, Web Services, along with many crucial software engineering activities. In the end, we are going to publish this application to Google's Play Store to make it easier for everyone to install it. We are also planning to release many new features in the future such as a Statistics page to show useful insights for the team leader and to improve the team.

10. REFERENCES

- [1]. IEEE Recommended Practice for Software Requirements Specifications. (1998).
Ieeexplore.ieee.org. Retrieved 27 December 2017, from
<http://ieeexplore.ieee.org/document/720574/>
- [2]. IEEE Recommended Practice for Software Design Descriptions. (1998).
Ieeexplore.ieee.org. Retrieved 27 December 2017, from
<http://ieeexplore.ieee.org/document/741934/>
- [3]. Project Management Software | Microsoft Project. (2016). Products.office.com.
Retrieved 27 December 2017, from <https://products.office.com/en-us/project/project-and-portfolio-management-software?tab=tabs-1>
- [4]. Modelio Open Source - UML and BPMN free modeling tool. (2017). Modelio.org.
Retrieved 27 December 2017, from <https://www.modelio.org/>
- [5]. Flowchart Maker & Diagramming Software, Microsoft Visio. (2016). Products.office.com. Retrieved 27 May 2017, from <https://products.office.com/en-us/visio/flowchart-software?tab=tabs-1>
- [6]. MockFlow - Online Wireframe and UX Tools. Mockflow.com. Retrieved 27 May 2017, from <https://www.mockflow.com/>
- [7]. Flowchart Maker & Online Diagram Software. Draw.io. Retrieved 27 May 2017, from <http://draw.io>
- [8]. SQL Server 2016 | Microsoft. Microsoft SQL Server - US (English). Retrieved 27 May 2017, from <https://www.microsoft.com/en-us/sql-server/sql-server-2016>
- [9]. Stroud, D. (2017). Visual Studio IDE, Code Editor, VSTS, & App Center. Visual Studio. Retrieved 27 December 2017, from <https://www.visualstudio.com/>
- [10]. Microsoft Azure: Cloud Computing Platform & Services. Azure.microsoft.com.
Retrieved 27 May 2017, from <http://azure.microsoft.com>
- [11]. JSON. Json.org. Retrieved 27 May 2017, from <http://www.json.org/>
- [12]. Design | Android Developers. (2017). Developer.android.com. Retrieved 27 December 2017, from <https://developer.android.com/design/index.html>

APPENDIX A

Instructions for Installing the system:

Copy the file “com.TeamTracker.calender-Signed.apk” on the CD accompanied with this report to any Android device with an Android Version of 4.4 or higher. On the Android device, locate the folder in which the file was copied, open the .apk file and click install.

APPENDIX B

Code for the system: 1LoginPage.xaml.cs

This is the C# code for the main login page of the program. The code for all the other parts of the program can be found on the CD attached with the report.

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

using System.Net.Http;

using System.Net;

using Calendar.Model;

namespace Calendar

{

    [XamlCompilation(XamlCompilationOptions.Compile)]

    public partial class LoginPage1 : ContentPage

    {

        private string Url =

"http://teamloggerapi20171216093012.azurewebsites.net/api/App_Login?email={0}&pass={

1}";

        private string Url2 = "http://teamloggerapi20171216093012.azurewebsites.net/api/Live";

        private string Url3 =

"http://teamloggerapi20171216093012.azurewebsites.net/api/GetName";
```

```

private string result;

private static Api_Connector connec = new Api_Connector();

private static CookieContainer cookieContainer = new CookieContainer();

private static HttpClientHandler clienthandler =

    new HttpClientHandler

    {

        AllowAutoRedirect = true,

        UseCookies = true,

        CookieContainer = cookieContainer

    };

public static HttpClient client = new HttpClient(clienthandler);

private bool _isBusy;

bool Disabled = false;

public static GlobalProperties Globalproperties = new GlobalProperties();

public LoginPage1()

{

    InitializeComponent();

    NavigationPage.SetHasNavigationBar(this, false);

    EntryEmail.IsEnabled = false;

    EntryPassword.IsEnabled = false;

    //Registration_Button.IsEnabled = true;

    //LogIn_Button.IsEnabled = true;

    // RetryButton.IsVisible = true;

```



```

    RetryLabel.IsVisible = true;

    EntryPassword.Text = EntryEmail.Text = "";

    RetryButton.Source =
ImageSource.FromResource("Calendar.Images.RetryButton.png");

    LOGOImage.Source = ImageSource.FromResource("Calendar.Images.LOGO.png");

    BackGroundImage.Source =
ImageSource.FromResource("Calendar.Images.Background5.jpg");

    LoginImage.Source =
ImageSource.FromResource("Calendar.Images.LoginButton1.png");

    RegisterImage.Source =
ImageSource.FromResource("Calendar.Images.RegisterButton1.png");

    InfoIcon.Source = ImageSource.FromResource("Calendar.Images.Info.png");
}

async void OnLoginClicked(object sender, EventArgs e)
{
    //await Navigation.PushAsync(new TeamsDashboardPage3());

    //return;

    if (_isBusy) return;

    if (Disabled == true)

        return;

    if (EntryEmail.Text.Equals("") || EntryPassword.Text.Equals(""))
    {

        await DisplayAlert("Error", "enter both email and password", "ok");

        return;
    }
}

```

```

    }

    if (!EntryEmail.Text.Contains("@"))

    {

        await DisplayAlert("Error", "incorrect email address", "ok");

        return;

    }

    if (EntryPassword.Text.Length < 6)

    {

        await DisplayAlert("Error", "password should be at least 6 charecters", "ok");

        return;

    }

    _isBusy = true;

    result = await Api_Connector.Connect(String.Format(Url, EntryEmail.Text,
EntryPassword.Text));

    switch (result.ToString())

    {

        case "0":

            await DisplayAlert("Error", "Wrong Username or Password", "ok");

            _isBusy = false;

            return;

        case "1":

            Globalproperties.UserName = await Api_Connector.Connect(Url3);

            await Navigation.PushAsync(new TeamsDashboardPage3());

```

```

        _isBusy = false;

        return;

    case "404":

        await DisplayAlert("system error", "System is down\n please try again later",
"ok");

        _isBusy = false;

        return;

    case "666":

        await DisplayAlert("system error", "input error", "ok");

        _isBusy = false;

        return;

    case "999":

        await DisplayAlert("communication error", "please check your internet
connection\nand try again", "ok");

        _isBusy = false;

        return;

    default:

        await DisplayAlert("Unknown Error", "Unknown Error", "ok");

        _isBusy = false;

        return;

    }

}

```

```

async void OnRegisterClicked(object sender, EventArgs e)
{
    if (_isBusy) return;

    if (Disabled == true)

        return;

    await Navigation.PushAsync(new RegistrationPage2());
}

```

```

protected override void OnAppearing()
{
    base.OnAppearing();

    successlabel.IsVisible = false;

    EntryPassword.Text = EntryEmail.Text = "";

    connection_check();
}

```

```

private async void connection_check()
{
    for (int i = 0; i < 3; i++)

    {

        result = await Api_Connector.Connect(Url2);

        if (result.ToString().Equals("111"))

        {

            EntryEmail.IsEnabled = true;

```

```

        EntryPassword.IsEnabled = true;

        if (Disabled == true)

        {

            Disabled = false;

            successlabel.IsVisible = true;

        }

        RetryButton.IsVisible = false;

        RetryLabel.IsVisible = false;

        return;

    }

}

EntryEmail.IsEnabled = false;

EntryPassword.IsEnabled = false;

Disabled = true;

RetryButton.IsVisible = true;

RetryLabel.IsVisible = true;

return;

}

async private void ToolbarItem_Clicked(object sender, EventArgs e)

{

    await DisplayAlert("About Us", "Application Developed by: \n*Mohamed Balto
\n*Abdoulgwad Elsheredi \n*Talal Mahdy \n*Adham Moshasha", "Close"); } }

}

```

APPENDIX C

Abbreviations and Definitions:

SDS: Software Design Specification Document

SRS: Software Requirements Specification Document

DB: Database

UML: Unified Modeling Language

TUBITAK: The Scientific and Technological Research Council of Turkey

BPMN: Business Process Modeling Notation

GUI: Graphical User Interface

API: Application Program Interface

UI: User Interface

UFP: Unadjusted Function Points

TCF: Technical Complexity Factor

FP: Function Points

EMU: Eastern Mediterranean University

IEEE: Institute of Electrical and Electronics Engineers

ERD: Entity Relationship Diagram

BPMN: Business Process Model Notation

IDE: Integrated Development Environment

SQL: Structured Query Language

MAU: Monthly Average Users

OS: Operating System

XAML: Extensible Application Markup Language

ASP: Active Server Pages

API: Application Programming Interface