A Major Project Report On

PROJECT TRACKER

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Engineering in Software Engineering at Pokhara University

By

BISHOWRAJ LAMICHHANE

HARI KRISHNA BHANDARI

SUJAN KUMAL

Department of Research and Development

GANDAKI COLLEGE OF ENGINEERING AND SCIENCE

Lamachaur, Kaski, Nepal

(November, 2018)

A Major Project Report On

PROJECT TRACKER

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Engineering in Software Engineering at Pokhara University

By

BISHOWRAJ LAMICHHANE

HARI KRISHNA BHANDARI

SUJAN KUMAL

Supervisor

ER. NISHAL GURUNG

Department of Research and Development

GANDAKI COLLEGE OF ENGINEERING AND SCIENCE

Lamachaur, Kaski, Nepal

(November, 2018)

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We extend our sincere thanks to all of them.

We express our gratitude to Project Supervisor Er. Nishal Gurung, Project Head Er. Sujan Tamrakar as well as our Vice Principal Mr. Ashok Raj Parajuli.

We are highly indebted to GCES for their guidance and constant supervision as well as for providing necessary information regarding the project and also for their support in completing the project.

We express our gratitude towards our parents and friends for their kind co-operation and encouragement which help us in completion of this project.

Our thanks and appreciations also go to our colleague in developing the project and people who have willingly helped us out with their abilities.

ABSTRACT

"ProjectTracker" provides information to track level of effort to project head, supervisors and students. It is a web-based application.

Project Tracker uses QR code as a tool for attendance process. Based on the task completion and attendance process, system assigns automatic marks. Total marks assigned by automatic marking system is 20 and rest 80 marks depends on stakeholder's, responsible for marking, analysis of each individual work.

LIST OF ABBREVIATIONS

AJAX	Asynchronous JavaScript And XML
API	Application Programming Interface
CLI	Command Line Interface
CSS	Cascading Style Sheets
ER	Entity Relationship
HTML	Hypertext Markup Language
HTTP	Hyper Text Transfer protocol
IoC	Inversion of Control
iOS	iPhone Operating System
JS	JavaScript
JSON	JavaScript Object Notation
MVC	Model View Controller
ORM	Object Relational mapping
PHP	Personal Home Page
PT	Project Tracker
QR	Quick Reference

REST	REpresentational State Transfer
SMTP	Simple Mail Transfer Protocol
SRS	Software Requirements Specification
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UI	User Interface
XAMPP	Cross-Platform (X) Apache MySQL PHP Perl
XML	eXtensible markup language

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SCREENSHOTS

Chapter 1

INTRODUCTION

1.1 BACKGROUND

Project tracking begins early in the project with planning and goes on till the completion of a project. It helps project manager monitor how the projects are progressing so that potential problems are identified in a timely manner and take corrective action. The key benefit is that project performance is measured regularly to identify variances from the project management plan so that project manager can make sure project is on track. So, we figured out a way to eliminate the traditional system with digital and simple way to track projects with our Web application called “Project Tracker”.

1.2 PROBLEM STATEMENT

We have been involved in software projects like Junior Quiz buzz, Belako Boli, GCES Scheduler, Online Cricket as a part in partial fulfillment of the requirements for the Degree of Bachelor of Engineering in Software Engineering. The main problem everyone faces in this process is proper tracking of the project. To track the progress and individual efforts, papers are being used. Managing papers, going through them is boring task. Similarly, most of the time when deadline is near project member somehow manage to complete the project but makes difficult to determine each member effort in just.

1.3 PROJECT OBJECTIVE

The objectives of this project are as follows.

1.3.1 Primary Objective

- To track projects held in our college.

1.3.2 Specific Objective

- To monitor the projects progress
    - records about events.
    - meetings via QR code[1]/ Bar code.
    - individual tasks (task division).
- To provide the notification about the events via email.
- To provide information to stakeholders that justify the level of effort required to complete the project(s).

1.4 IMPLICATION

Project management nowadays is regarded as a high priority this is because organizations, whether small or large, are at one time or another involved in implementing new undertakings, innovation and changes on project. So, in completion and success of this project one will be able to see useful outcome.

---

[1] Quick Reference code is the trademark for a type of matrix barcode (or two-dimensional barcode) first designed in 1994 for the automotive industry in Japan.

Chapter 2

LITERATURE REVIEW

After some researching on this subject matter, we found similar project made on the same field. We present some projects that we researched during our study.

2.1 TRELLO

Trello is a web-based project management originally made by Fog Creek Software in 2011. Trello has a variety of work and personal uses including real estate management, software project management, school bulletin boards, lesson planning, accounting, web design, gaming and law office case management.

Trello's apps include:

- List tasks and ideas on cards
- Organize cards into lists and Kanban boards[2]
- Drag and drop cards to move between columns and create your own workflow
- Collaborate with teammates on shared cards
- Add power-ups to connect boards with 3rd party apps and automations
- iOS and Android apps available

[2] A Kanban board is a work and workflow visualization tool that enables people to optimize the flow of work. Physical Kanban boards typically uses sticky notes on a whiteboard to communicate status, progress, and issues.

2.2 FREEDCAMP

Freedcamp is a web-based project management tool and organization system for single or multiple users collaborating using cloud computing. The company was launched in 2010 in Santa Barbara, California by Angel Grablev.

Freedcamp's apps include:

- Tasks - Shared task lists with subtasks
- Discussions - Forum boards
- File Storage - Advanced file management
- Milestones - Deadline setting
- Calendar - Event scheduling
- Time - Time tracking
- Password Manager - Securely storing passwords
- Tasky - Private task list
- Issue Tracker - Advanced workflow issue management
- Invoices - Bill clients with invoices
- Wiki - Create internal or public documentation

On comparing our project with these applications, we came to conclude with the following:

Application / Features	TRELLO (Trello, 2011)	Freedcamp (Freedcamp, 2010)	Project Tracker

Ease of Use	Medium	Low	High
Understanding	Medium	Medium	High
Internet Connections	Required	Required	Required
Platform	Web and Android	Web and Android	Web
Progress Represent	Progress meter	Pie Chart	Pie Chart
Type	Android	Web	Web
Use in	Personal and Small-Medium Organization	Personal, Small-Medium Organization Large Enterprises	Personal and Small-Medium Organization
Notification	Yes	Yes	Yes
Paid	Yes	Yes	No
QR Code	No	No	Yes

Table 2.1 Comparison Table

Chapter 3

TOOLS AND METHODOLOGY

The primary features of the project can be outlined as:

- Tasks division to team member based on their interest.
- Weekly meeting to assess and propose restructuring of plans when required.
- Project documentation by team member(s) at the end of each task.
- Regular discussion with the project to update our supervisor about the progress of our project.

3.1 TOOLS

3.1.1 Laravel

Laravel a free and open source[3] PHP web framework[4] is used based on the scalability and features it offers. It is method for the development of web development applications following the model-view-controller architectural pattern (MVC)[5] . Laravel is web application PHP framework with advanced query syntax that makes web development simple and rapid

[3] Open-source software is a type of computer software in which source code is released under a license in which the copyright holder grants users the rights to study, change, and distribute the software to anyone and for any purpose.

[4] a **framework** is often a layered structure indicating what kind of programs can or should be built and how they would interrelate.

[5] The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application.

by enabling general tasks that will be used in the majority of web projects such as route, queue, sessions and authentication.

The foundation of our project is built upon MVC model, and today's date Laravel is one of the most popular PHP frameworks whose main focus is on MVC design pattern.

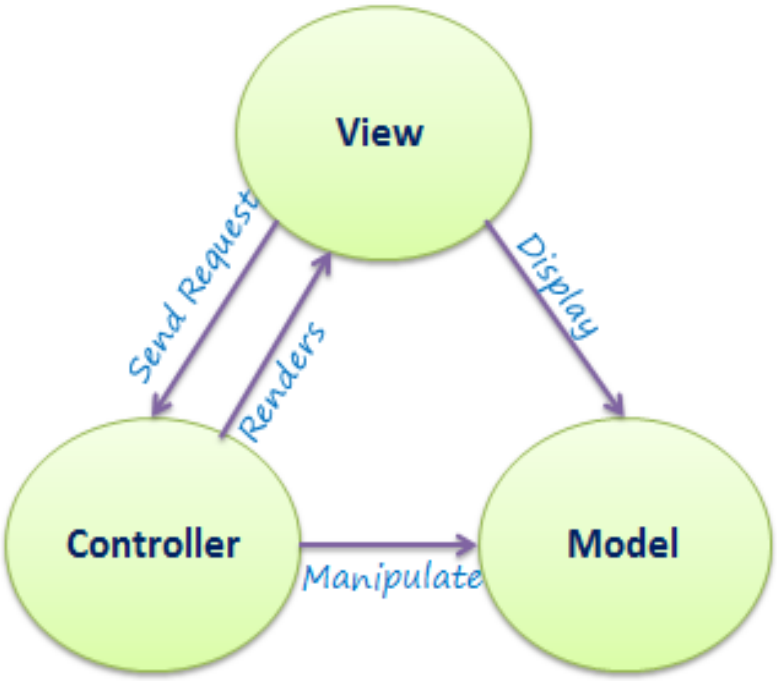


Figure 3.1 MVC Architecture

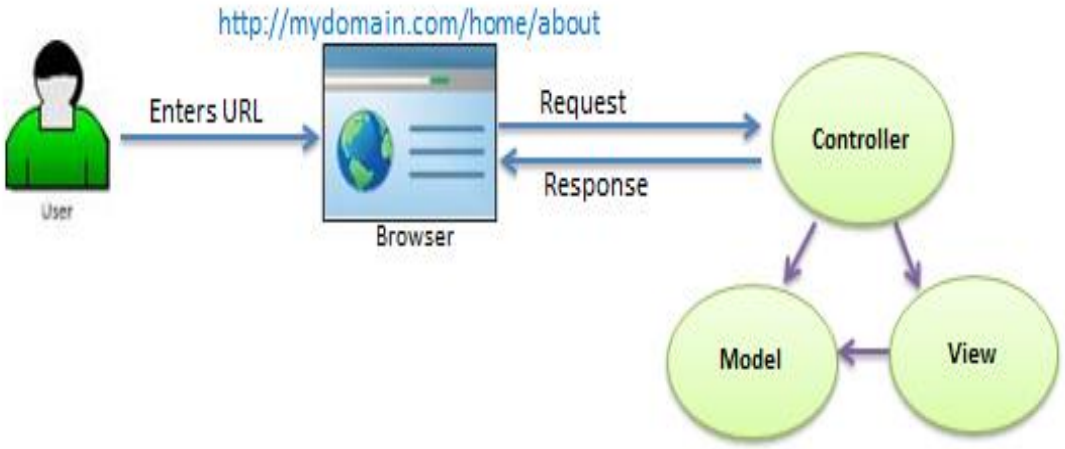


Figure 3.2 Request/Response in MVC Architecture

With the old approach, backend developers are expected to know much more about the UI and visual aspects of the application. Because of this, they have to pay attention to different segments of the application, instead of focusing on their primarily objective. Having the backed API strictly separated from the UI allowed us to focus on the quality of our code. Features of Laravel like routes, ORM easy implementation of REST API, separation of view and controller completely, blade engines for view helped a lot to build project without considering any other requirements.

3.1.2 MySQL

MySQL is an open source rational database management system (RDBMS). Its name is a combination of "My", the name of co-founder

Michael Wideness' daughter[6], and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL is offered under two different editions: the open source MySQL Community Server and the proprietary Enterprise Server. MySQL Enterprise Server is differentiated by a series of proprietary extensions which install as server plugins, but otherwise shares the version numbering system and is built from the same code base.

3.1.3 Visual Studio Code

Visual Studio Code is a source code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control, syntax highlighting, intelligent code completion, snippets, and code refactoring. It is also customizable, so users can change the editor's theme, keyboard shortcuts, and preferences. It is free and open-source, although the official download is under a proprietary license. It supports a number of programming languages and a set of features that may or may not be available for a given language. Many of Visual Studio Code features are not exposed through menus or the user interface. Rather, they are accessed via the command palette or via a .json file e.g., user preferences.

[6] The name MySQL owes its name to Monty's daughter My. Indeed Michael, often called Monty, has a habit of naming his projects after his children: MariaDB was named after his youngest daughter and MaxDB was named after his son Max.

3.1.4 Bootstrap

Bootstrap is a free and open-source front-end framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many earlier web frameworks, it concerns itself with front-end development only. Bootstrap supports responsive web design which lets the layout of web pages adjusts dynamically, considering the characteristics of the device used (desktop, tablet, mobile phone).

3.1.5 JavaScript

JavaScript often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm. Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web applications. The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it.

3.1.6 XAMPP

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning

from a local test server to a live server possible. XAMPP is regularly updated to the latest releases of Apache, MariaDB, PHP and Perl. It also comes with a number of other modules including OpenSSL, phpMyAdmin etc.

3.2 TECHNIQUES

3.2.1 Web Development Skills

A web developer should be able to:

- Wireframe/layout a website or web application.
- Add interactivity to a website using JavaScript.

3.2.2 Object Oriented Concept

- Project Tracker uses object-oriented paradigm.

3.3 DEVELOPMENT

Target Platform	Web
Programming Language	HTML, CSS, JavaScript, PHP
Distribution Method	Public Web Portal
Plans	6 Months
Plan Total Developer Cost	N/A
Total Developers	3

Table 3.1 Table of Tools

3.4 SOFTWARE DEVELOPMENT LIFECYCLE

The SDLC model we followed in developing this project is incremental model, which is a use of linear sequential model in an iterative manner. New functionalities were added as each increment was developed. Linear sequential model was applied to develop each increment. The phases of the linear sequential model are: Analysis, Design, Coding and Testing. The software repeatedly passes through these phases in iteration and an increment is delivered with progressive changes.

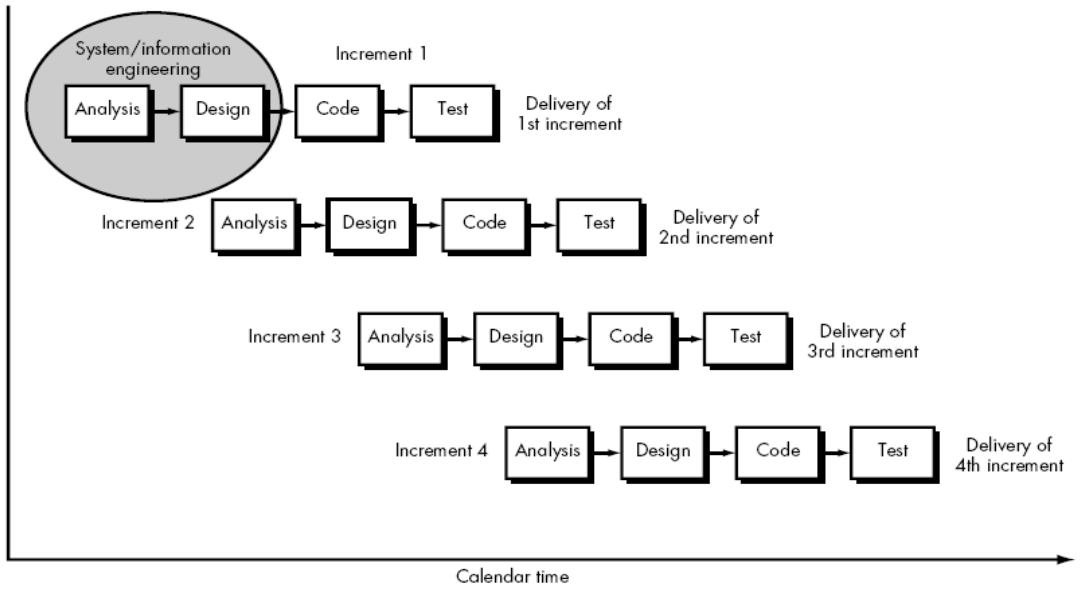


Figure 3.3 Incremental Model

Analysis Phase: In this phase, analysis was done in order to find out the requirements of the system.

Design Phase: In this phase the SRS was translated into the system's design. Use Case Diagram and Class Diagram were developed.

Coding Phase: In this phase coding was done according to the design and a working system was developed by the end of this process.

Testing Phase: In this phase, the system was tested. With each testing a list of changes to the system developed, was suggested and the changes were applied to the software.

Chapter 4

SYSTEM ANALYSIS AND DESIGN

4.1 USE CASE DIAGRAM

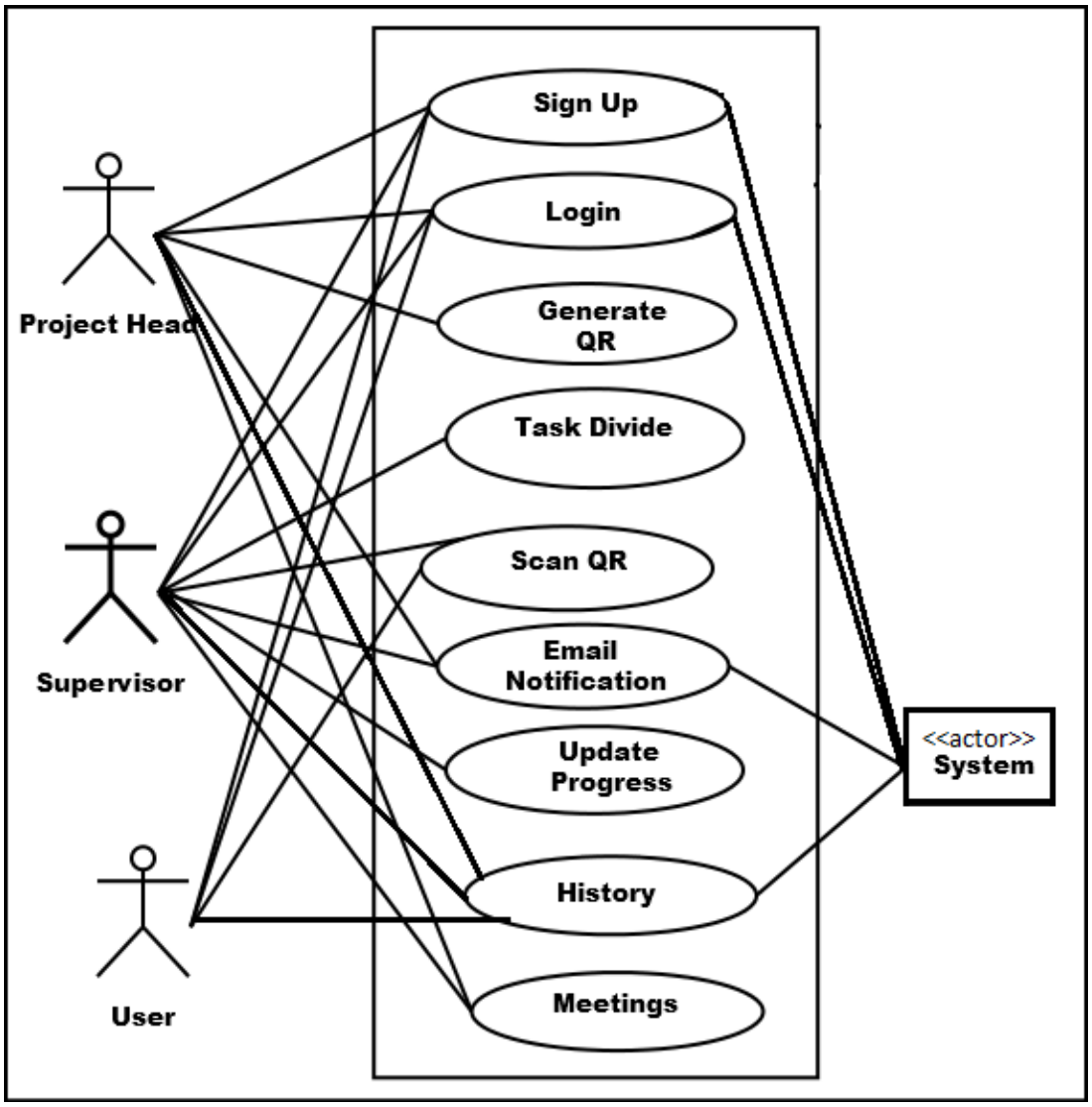


Figure 4.1 Usecase diagram of "ProjectTracker"

The Use Case descriptioin for above use case are depicted below:

Use Case 1:	Signup
Primary Actor:	User
Scenario:	Signup the user into the application.
Stakeholders and Interest:	User: Wants to access the application.
Brief Description:	For users to access the system, they have to fill up the given form field and only after the verification they have their respective username and password to access the application.
Preconditions:	Email must be accessible.
Postconditions:	User is identified and verified.
Main Success Scenario:	User gets access to the application.

Use Case 2:	Login
Primary Actor:	User
Scenario:	Sign-in the user into the application.
Stakeholders and Interest:	User: Wants to access application.
Brief Description:	User enters username and password and clicks on login button. The system verifies the username and password and if correct, logs the user into the application else it shows invalid username/password. Or user login via google social login service.
Preconditions:	User should have an account. And for google login user should have google account.
Postconditions:	Should display the user's dashboard after successfully logged into the system.

	If logged in via google, password must be set.
Main Success Scenario:	After the authentication, the user's dashboard is displayed to the user.
Exception Conditions:	1. If user doesn't exist, then they can create an account. 2. If the user forgets the username/password then they can reset username/password.

Use Case 3:	Generate QR
Primary Actor:	User
Scenario:	The user generates QR for the respective project.
Brief Description:	User selects the project and generate QR.
Preconditions:	1. User must be project head. 2. User should be signed into the application and must be project head of the selected project.

Postconditions:	Generate QR.
Main Success Scenario:	Display QR code.

Use Case 4:	Task
Primary Actor:	User
Scenario:	The user adds Task.
Preconditions:	1: User should be signed-in on the application and should be Leader or Supervisor. 2: Project must be selected.
Postconditions:	Task added to the project.
Main Success Scenario:	Task must be added and displayed in task view.
Exception Conditions:	If the item is not available then the user can request the item.

Use Case 5:	Scan QR
Primary Actor:	User
Scenario:	The user scans the QR code.
Stakeholder and Interest:	User: Wants to Scan the QR code.
Preconditions:	1: User should be signed-in on the application and should be supervisor, leader or member. 2: QR code must be available.
Postconditions:	Attendence is done.
Main Success Scenario:	Attendence is done and success message is displayed.
Exception Conditions:	If QR is invalid Error is displayed.

Use Case 6:	Email Notification
Primary Actor:	User

Scenario:	The user sends notification.
Stakeholder and Interest:	User: Wants to send notification.
Preconditions:	1: User should be signed-in on the application. 2: Project must be selected.
Postconditions:	System is updated with notification send.
Main Success Scenario:	Members of the selected project get email notification

Use Case 7:	Meeting minute
Primary Actor:	User
Scenario:	The user fills minute form.
Stakeholder and Interest:	User: Wants to create minute.

Preconditions:	1: User should be signed-in on the application and must be supervisor or leader. 2: Task must be available
Postconditions:	Minute created.
Main Success Scenario:	Minute created message displayed.

4.2 CLASS DIAGRAM

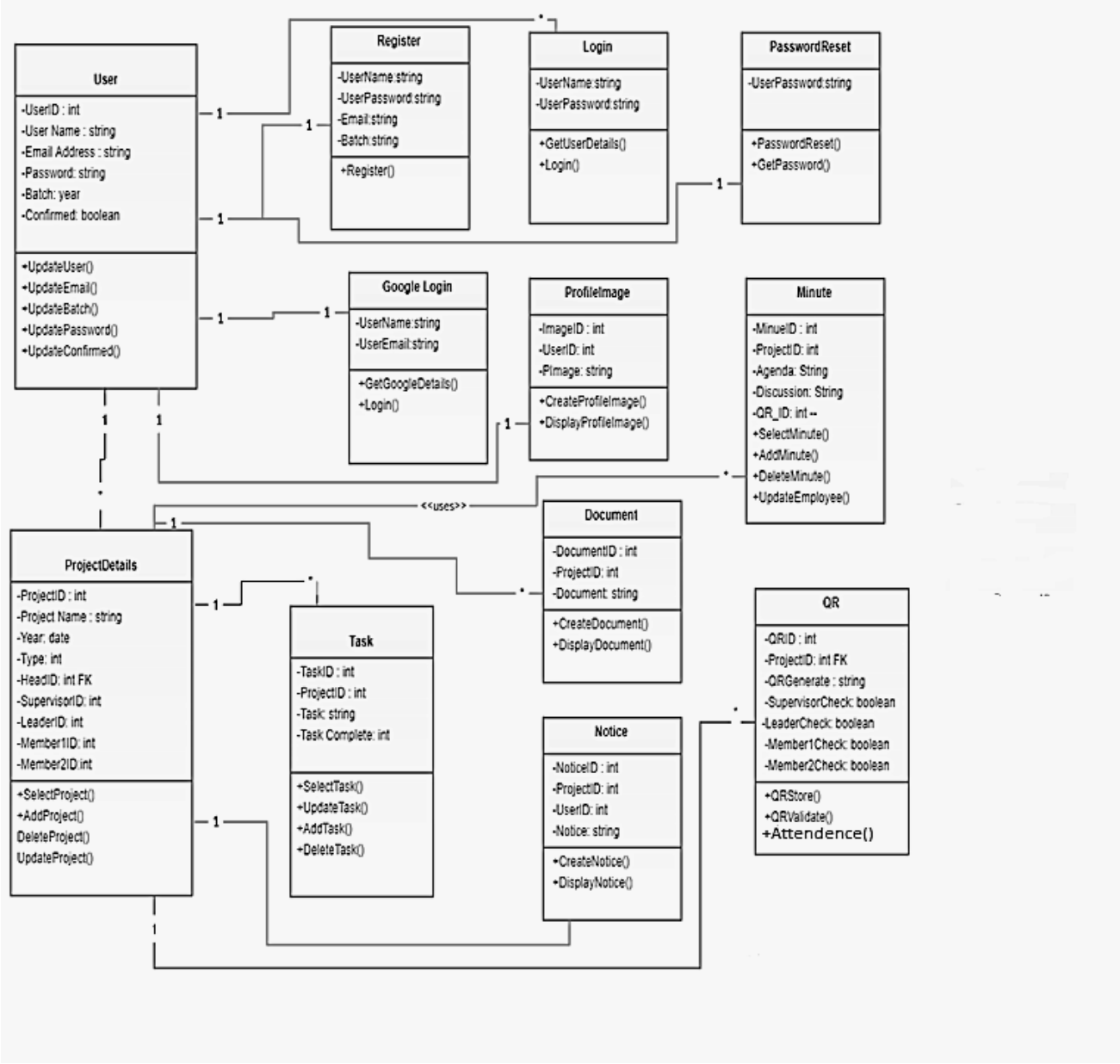


Figure 4.2 Class Diagram

4.3 SEQUENCE DIAGRAM

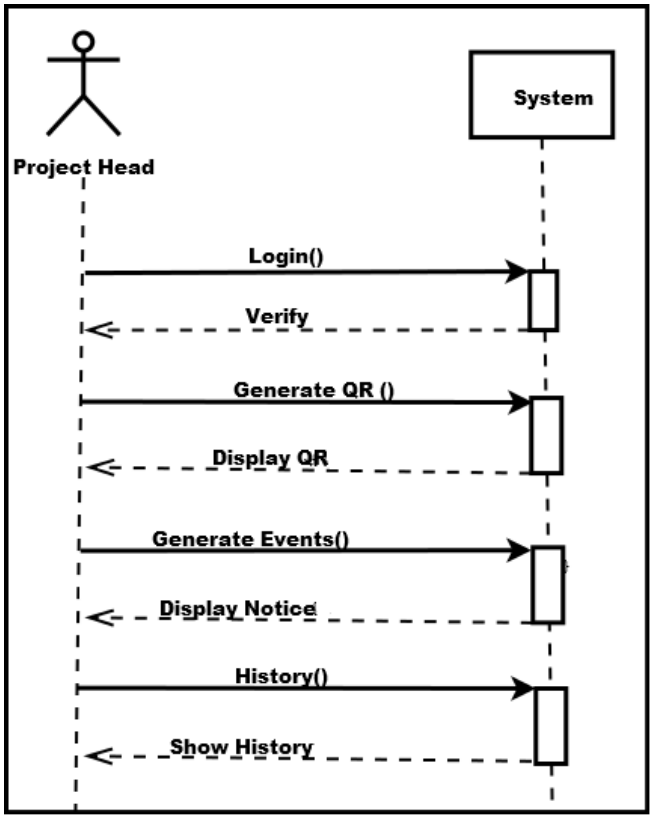


Figure 4.3 System Sequence Diagram (Admin-System)

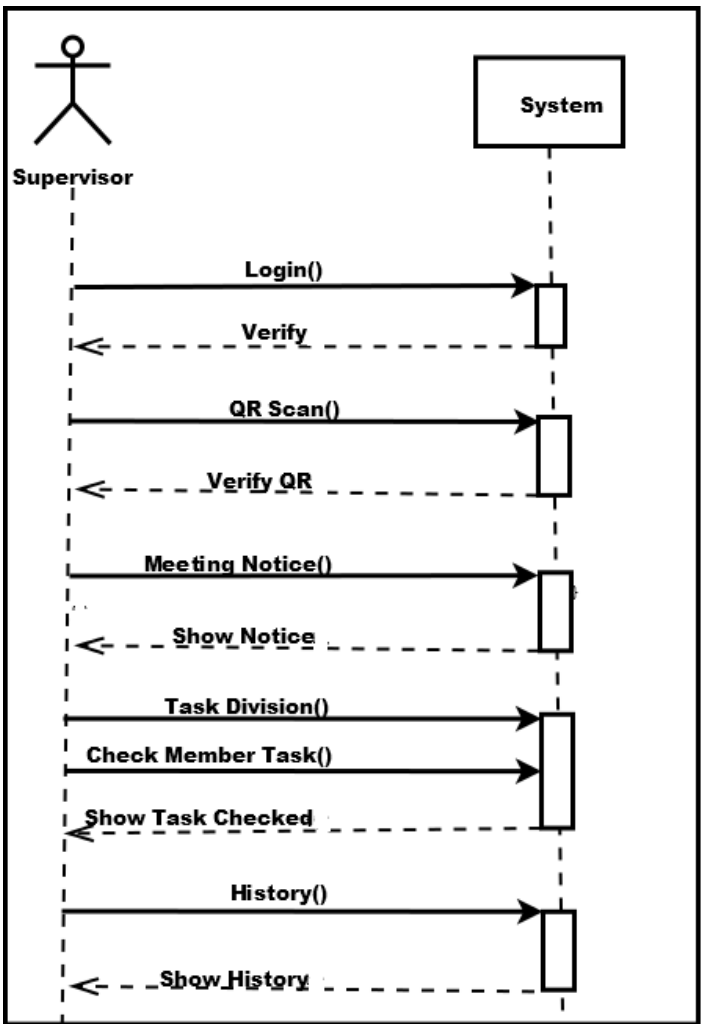


Figure 4.4 System Sequence Diagram (Supervisor-System)

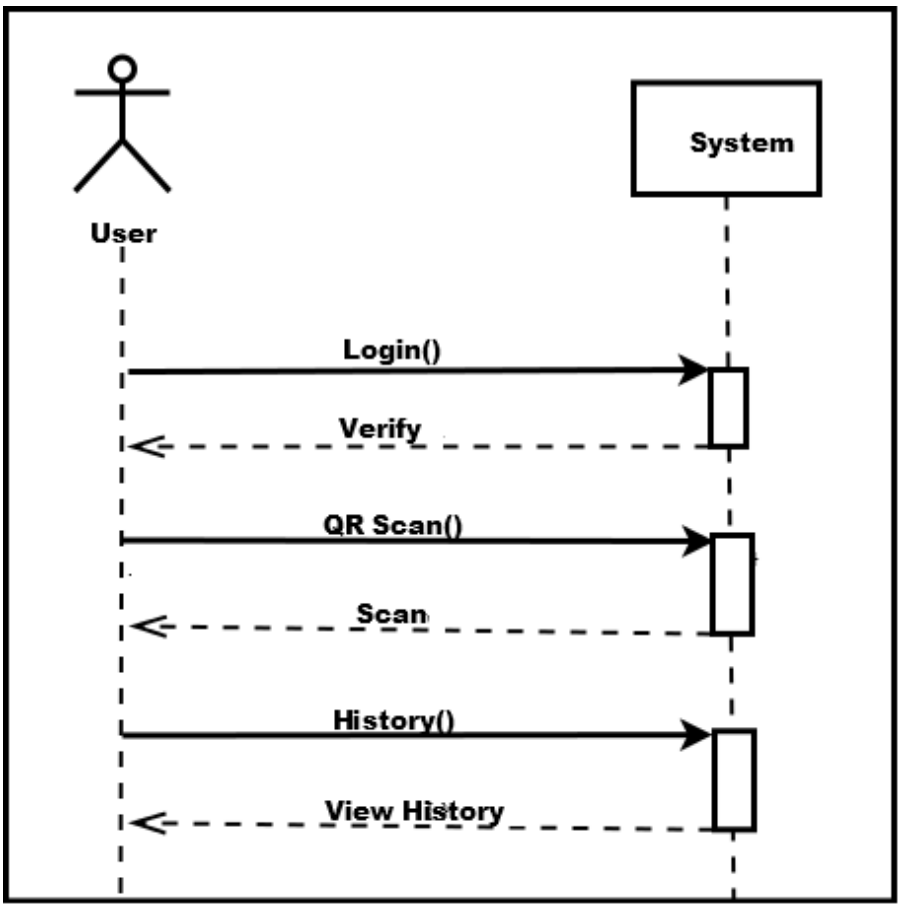


Figure 4.5 System Sequence Diagram (User-System)

4.4 DATABASE DIAGRAM

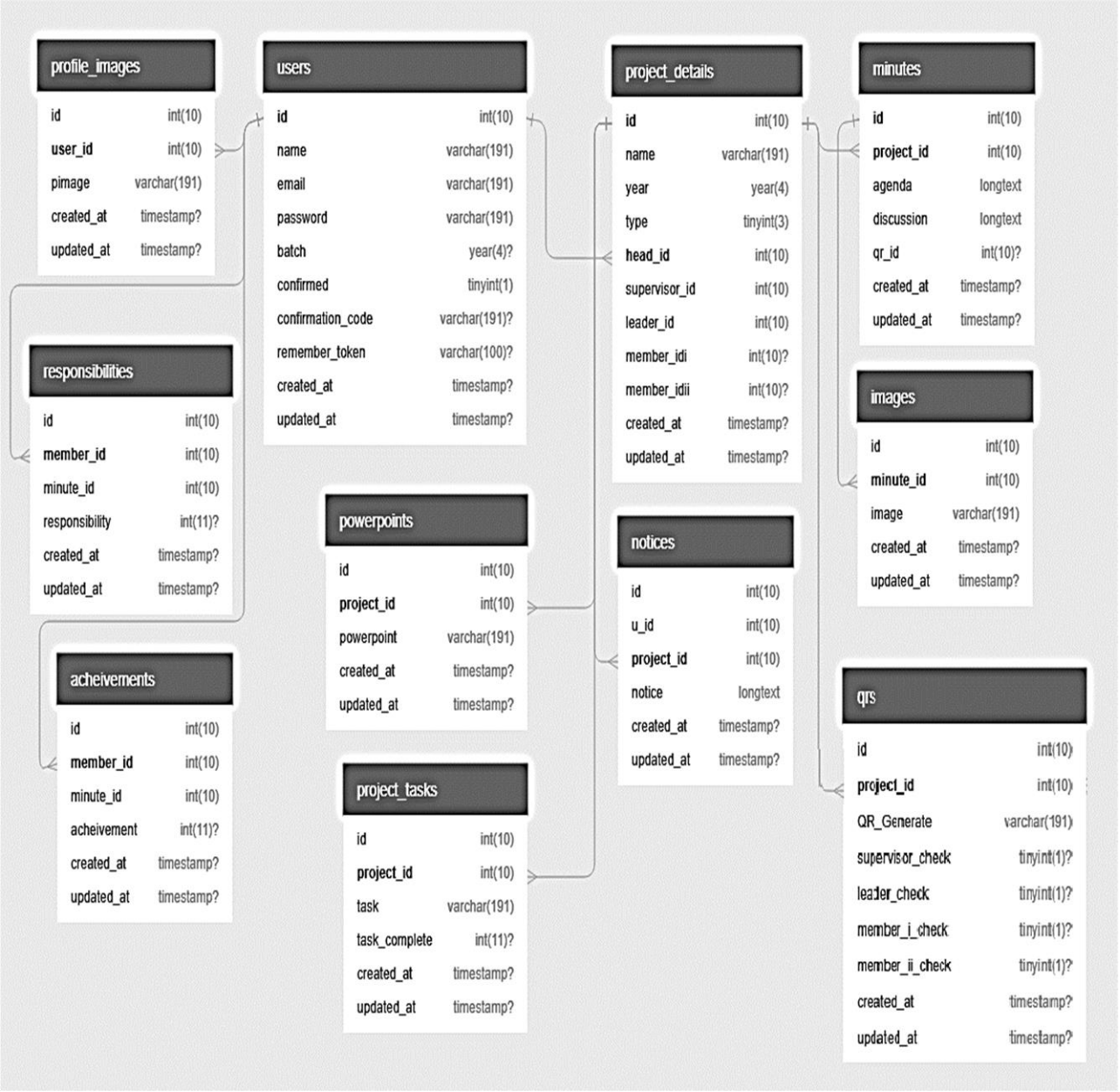


Figure 4.6 Database diagram

4.5 ER DIAGRAM

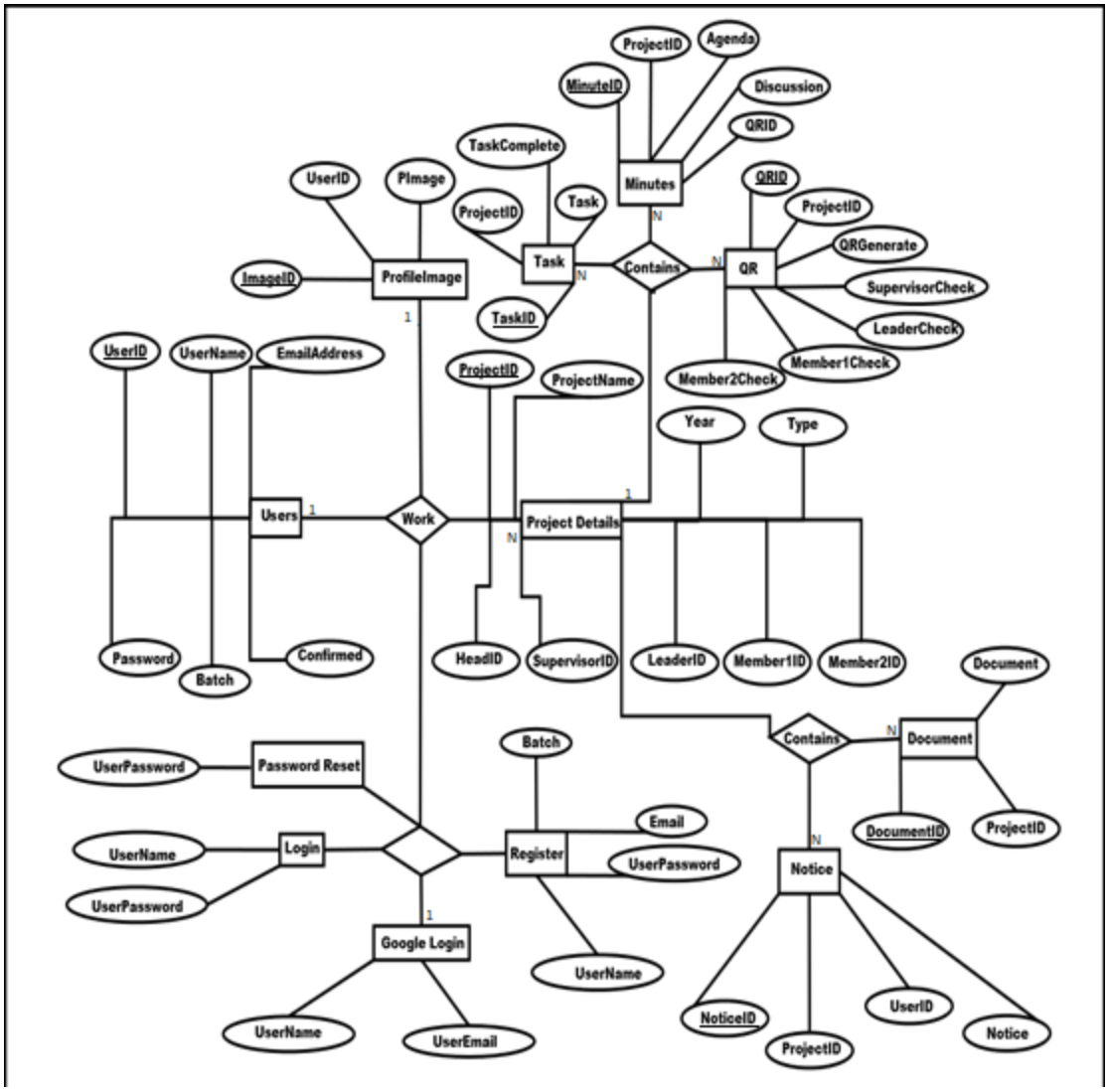


Figure 4.7 ER diagram

4.6 TRAFFIC FLOW DIAGRAM FOR EMAIL

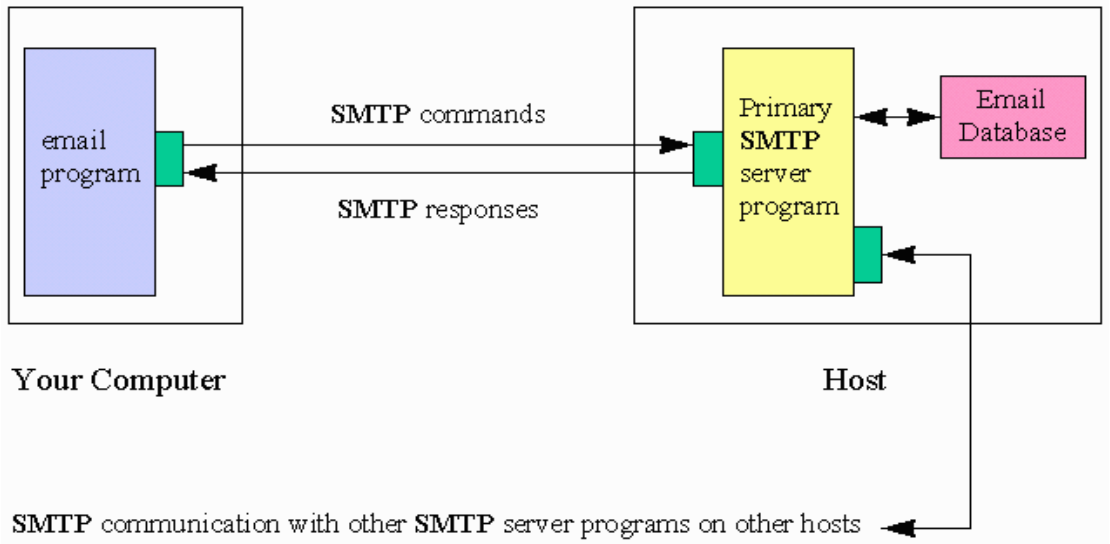


Figure 4.8 Traffic flow diagram of Email (1)

4.7 FLOW DIAGRAM FOR QR

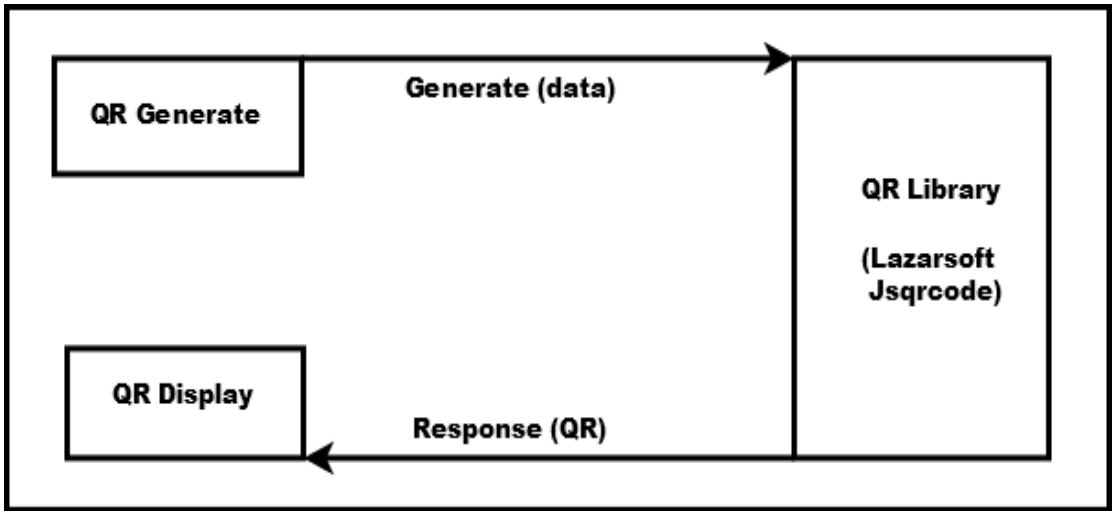


Figure 4.9 Flow diagram to Generate QR

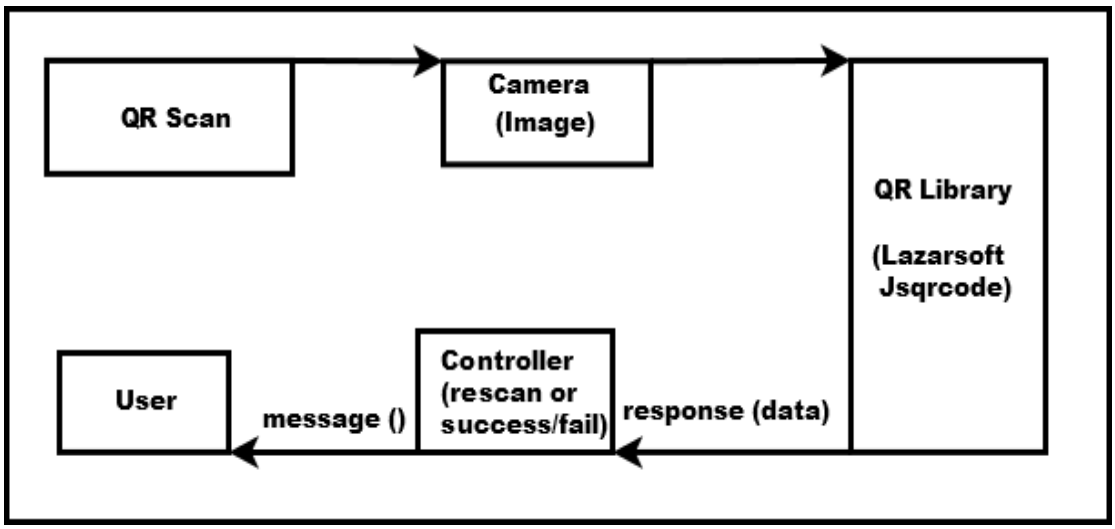


Figure 4.10 Flow diagram to Scan QR

4.8 DATA DICTIONARY

4.8.1 Achievements

	Name	Data type	Description / Attributes
🔑	Id	unsigned int(10, 0)	Identity / Auto increment
	member_id	unsigned int(10, 0)	References: users
	minute_id	unsigned int(10, 0)	References: minutes
	Achievement	int(10, 0)	Nullable
	created_at	Timestamp	Nullable
	updated_at	Timestamp	Nullable

Table 4.1 Achievements

4.8.2 Images

	Name	Data type	Description / Attributes
🔑	Id	unsigned int(10, 0)	Identity / Auto increment
	minute_id	unsigned int(10, 0)	References: minutes
	Image	varchar(30)	
	created_at	Timestamp	Nullable
	updated_at	Timestamp	Nullable

Table 4.2 Images

4.8.3 Migrations

	Name	Data type	Description / Attributes
🔑	id	unsigned int(10, 0)	Identity / Auto increment
	migration	varchar(191)	
	batch	int(10, 0)	

Table 4.3 Migrations

4.8.4 Minutes

	Name	Data type	Description / Attributes
🔑	id	unsigned int(10, 0)	Identity / Auto increment
	project_id	unsigned int(10, 0)	References: project_details
	agenda	Longtext	
	discussion	Longtext	

	qr_id	unsigned int(10, 0)	Nullable References: qrs
	created_at	Timestamp	Nullable
	updated_at	Timestamp	Nullable

Table 4.4 Minutes

4.8.5 Notices

	Name	Data type	Description / Attributes
🔑	Id	unsigned int(10, 0)	Identity / Auto increment
	u_id	unsigned int(10, 0)	References: users
	project_id	unsigned int(10, 0)	References: project_details
	Notice	Longtext	
	created_at	Timestamp	Nullable
	updated_at	Timestamp	Nullable

Table 4.5 Notices

4.8.6 PowerPoint

	Name	Data type	Description / Attributes
🔑	id	unsigned int(10, 0)	Identity / Auto increment
	project_id	unsigned int(10, 0)	References: project_details
	powerpoint	varchar(30)	

	Name	Data type	Description / Attributes
🔑	Id	unsigned int(10, 0)	Identity / Auto increment
	user_id	unsigned int(10, 0)	References: users
	pimage	varchar(30)	
	created_at	Timestamp	Nullable
	created_at	Timestamp	Nullable
	updated_at	Timestamp	Nullable

Table 4.6 PowerPoint

4.8.7 Profile images

Table 4.7Profile images

4.8.8 Project details

	Name	Data type	Description / Attributes
🔑	id	unsigned int(10, 0)	Identity / Auto increment
	name	varchar(191)	
	year	Year	
	type	unsigned tinyint(3, 0)	

	head_id	unsigned int(10, 0)	References: users
	supervisor_id	unsigned int(10, 0)	References: users
	leader_id	unsigned int(10, 0)	References: users
	member_idi	unsigned int(10, 0)	Nullable References: users
	member_idii	unsigned int(10, 0)	Nullable References: users
	created_at	Timestamp	Nullable
	updated_at	Timestamp	Nullable

Table 4.8 Project Details

4.8.9 Project tasks

	Name	Data type	Description / Attributes
🔑	id	unsigned int(10, 0)	Identity / Auto increment
	project_id	unsigned int(10, 0)	References: project_details
	task	varchar(191)	
	task_complete	int(10, 0)	Nullable
	created_at	Timestamp	Nullable
	updated_at	Timestamp	Nullable

Table 4.9 Project Tasks

4.8.10 QRs

	Name	Data type	Description / Attributes
🔑	Id	unsigned int(10, 0)	Identity / Auto increment
	project_id	unsigned int(10, 0)	References: project_details
	QR_Generate	varchar(60)	
	supervisor_check	tinyint(3, 0)	Nullable
	leader_check	tinyint(3, 0)	Nullable
	member_i_check	tinyint(3, 0)	Nullable
	member_ii_check	tinyint(3, 0)	Nullable
	created_at	Timestamp	Nullable
	updated_at	Timestamp	Nullable

Table 4.10 QRs

4.8.11 Responsibilities

	Name	Data type	Description / Attributes
🔑	id	unsigned int(10, 0)	Identity / Auto increment
	member_id	unsigned int(10, 0)	References: users
	minute_id	unsigned int(10, 0)	References: minutes
	responsibility	int(10, 0)	Nullable
	created_at	Timestamp	Nullable
	updated_at	Timestamp	Nullable

Table 4.11 Responsibilities

4.8.12 Users

	Name	Data type	Description / Attributes
	Id	unsigned int(10, 0)	Identity / Auto increment
	Name	varchar(30)	
	Email	varchar(30)	
	Password	varchar(191)	
	Batch	Year	Nullable
	Confirmed	tinyint(3, 0)	Default: 0
	confirmation_code	varchar(100)	Nullable
	remember_token	varchar(100)	Nullable
	created_at	Timestamp	Nullable
	updated_at	Timestamp	Nullable

Table 4.12 Users

4.8.13 Verify users

	Name	Data type	Description / Attributes
	id	unsigned int(10, 0)	Identity / Auto increment
	user_id	int(10, 0)	
	token	varchar(191)	
	created_at	Timestamp	Nullable
	updated_at	Timestamp	Nullable

Table 4.13 Verify Users

4.8.14 Marks

	Name	Data type	Description / Attributes
	Id	unsigned int(10, 0)	Identity / Auto increment
	project_id	unsigned int(10, 0)	References: project_details
	pd_leader	int(10, 0)	Nullable
	pd_mem1	int(10, 0)	Nullable
	pd_mem2	int(10, 0)	Nullable
	fd_leader	int(10, 0)	Nullable
	fd_mem1	int(10, 0)	Nullable
	fd_mem2	int(10, 0)	Nullable
	doc_leader	int(10, 0)	Nullable
	doc_mem1	int(10, 0)	Nullable
	doc_mem2	int(10, 0)	Nullable
	tw_leader	int(10, 0)	Nullable
	tw_mem1	int(10, 0)	Nullable
	tw_mem2	int(10, 0)	Nullable
	ss_leader	int(10, 0)	Nullable
	ss_mem1	int(10, 0)	Nullable
	ss_mem2	int(10, 0)	Nullable
	fm_leader	int(10, 0)	Nullable
	fm_mem1	int(10, 0)	Nullable
	fm_mem2	int(10, 0)	Nullable
	created_at	Timestamp	Nullable
	updated_at	Timestamp	Nullable

Table 4.14 Marks

Chapter 5
IMPLEMENTATION

The project “Project Tracker” is developed on top of PHP web framework Laravel version 5.6.4 created by Taylor Otwell and intended for the development of web applications following the model-view-controller (MVC) architectural pattern. Laravel is referred to as a “full stack” framework because it handles everything from the web serving to database management right down to HTML generation. A vertically integrated web development environment can provide a better experience for the developer. Laravel’s templating engine “Blade” did made our development much easier and hassle free.

The typical developer interacts with Laravel through the command line utility that generates and merges the Laravel project environment. Laravel comes with an excellent command line tool named Artisan that can be used to generate skeleton code and database schema stubs. Artisan handles everything from database schema migration to asset and configuration management. Below are the major reason that made us choose Laravel for our project.

- **Bundles** provides modular packaging system since the release of Laravel 3, with bundled features already available for easy addition to applications. Furthermore, Laravel 4 uses Composer (Nils Adermann, 2012) as a dependency manager to add framework agnostic and Laravel specific PHP packages.

- **Eloquent ORM** (object-relational mapping) is an advanced PHP implementation of the active record pattern, providing at he same time internal methods for enforcing constraints on the relationships between database objects. Following the active record pattern, Eloquent ORM presents database tables as classes, with their object instances tied to single table rows.
- **Query builder**, available since Laravel 4, provides a more direct database access alternative to Eloquent ORM. Instead of requiring SQL queries to be written directly, Laravel's query builder provides a set of classes and methods capable of building queries programmatically. It allows selectable caching of the results of executed queries.
- **Application logic** is an integral part of developed applications, implemented either by using controllers or as part of route declarations. The syntax used to define application logic is similar to the one used by Sinatra framework.
- **Reverse routing** defines a relationship between the links and routes, making it possible for later changes to routes to be automatically propagated into relevant links. When the links are created by using names of existing rotes thee appropriates uniform resource identifiers (URLs) are automatically created by Laravel.
- **Restful controllers** provide an optional way for separating the logic behind serving HTTP GET and POST requests.
- **Class auto loading** provides automated loading of PHP classes without the need for manual maintenance of inclusion paths. On

demand loading prevents inclusion of unnecessary components, so only the actually used components are loaded.

- **View composers** serve as customizable logical code units that can be executed when a view is loaded.
- **Blade templating engine** combines one or more templates with a data model to produce resulting views, doing that by transpiring the templates into the cached PHP code for improved performance. Blade also provides a set of its own control structures such as conditional statements and loops, which are internally mapped to their PHP counterparts. Furthermore, Laravel services may be called from Blade templates, and the templating engine itself can be extended with custom directives.
- **IoC containers** make it possible for new objects to be generated by following the inversion of control (IoC) principle, in which the framework calls into the application or task specific code, with optional instantiating and referencing of new objects as singletons.
- **Migrations** provide a version control system for database schemas, making it possible to associate changes in the application's code base and required changes in the database layout. As a result, this feature simplifies the deployment and updating of Laravel based applications.
- **Database seeding** provides a way to populate database tables with selected default data that can be used for applications testing or be performed as part of the initial application setup.
- **Unit testing** is provided as an integral part of Laravel, which itself contains unit tests that detect and prevent regressions in the

framework. Unit tests can be run through the provided artisan command line utility.

- **Automatic pagination** simplifies the task of implementation pagination, replacing the usual manual implementation approaches with automated methods integrated into Laravel.
- **Form request** is a feature of Laravel 5 that serves as the base for form input validation by internally binding event listeners, resulting in automated invoking of the form validation methods and generation of the actual form.
- **Filesystem,** introduced in Laravel 5.0, is a file system abstraction layer that allows local file systems and cloud-based storage services.
- **Artisan CLI,** a Laravel's command-line interface (CLI), called Artisan, was initially introduced in Laravel 3 with limited set of capabilities. Laravel's later migration to a Composer based architecture allowed Artisan to incorporate different components from the Symphony (Symphony, 2018) framework, resulting in the availability of additional Artisan features in Laravel 4.

  The features of Artisan are mapped to different subcommands of the artisan command-line utility, providing functionality that aids in managing and building Laravel-based applications. Common uses of Artisan include managing database migrations and seeding, publishing package assets, and generating boilerplate code for new controllers and migrations, the latter frees the developer from creating proper code skeletons. The functionality and capabilities of Artisan can also be expanded by implementing new custom

commands, which for example, may be used to automate application-specific recurring tasks.

5.1 ENVIRONMENT CONFIGURATION

The environment configuration is Laravel is a lot easier. The framework provides an env file (.env stands for environment) which stores all of the environment variables of the system. We have tried to write self-documented code as much as possible which provides readability and understandability among team members and other. The following code snippet shows the .env file of our project:

```
APP_NAME=ProjectTracker
APP_ENV=local
APP_KEY=base64:bRpbmmxyK3CQr22SR8Q+SFdqZe0TMyk6ZqPhpr3hFPg=
APP_DEBUG=true
APP_LOG_LEVEL=debug
APP_URL=http://localhost

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=projecttracker
DB_USERNAME=root
DB_PASSWORD=

BROADCAST_DRIVER=log
CACHE_DRIVER=file
SESSION_DRIVER=file
QUEUE_DRIVER=sync

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_DRIVER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=306e0dd40b2fa9
```

Figure 5.1 Environment Configuration

5.2 COMPONENT OF LARAVEL

Since Laravel follows model-view-controller (MVC) architectural pattern, the whole project application is divided as shown in the diagram below:

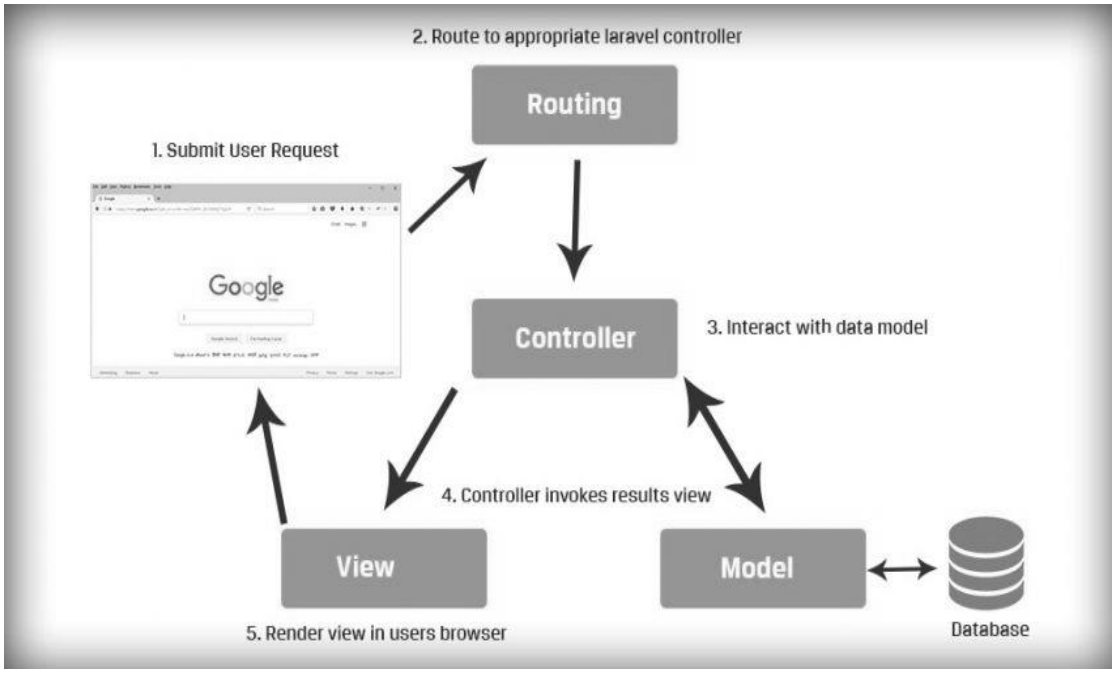


Figure 5.2 Component of Laravel

5.3 FILE STRUCTURE

Root Folder	Purposes
app	It contains models and controllers of the application. It also contains all of the logic for performing any operation. Models are placed directly inside this folder.
bootstrap	This folder contains the basic setting for starting the application.

config	This folder contains all configuration settings of the application like database connection, including core classes, email setting etc.
database	This folder contains the code for database transactions migrations, which are responsible for executing MySQL codes and Seeder class for testing with dummy data.
public	This folder can be seen by outer world. This is the directory that is to be pointed to web server. All static assets (like CSS, JavaScript, etc.) are placed here. The index.php file in this folder calls the bootstrap folder files and other core files and starts the application. The index.php receives all the requests and after complex process responses the content to the visitor. In other words, this is the file which transactions with browser.
resources	This folder contains the resources of the application like views, raw assets, language for localization, etc. Here the blade engine is implemented, and all the views are written by utilizing the blade structure.
routes	The most basic Laravel routes accept a URI and a Closure, providing a very simple and expressive method of defining routes.
storage	This folder stores the local data like sessions, caches, compiled views files, etc.
Tests	This folder contains all the tests of application.

vendor	This folder contains all the third-party files (dependencies and additional prepackages for plugins) and code files of the Laravel frameworks.

Table 5.1 File Structure Table

Chapter 6

TESTING

6.1 SCOPE

The overall purpose of testing was to ensure that Project Tracker web application meets all its technical and functional requirements. The purpose of this document is to describe the overall testing done for our applications. The approach described in this document provides the framework for all testing related to these applications.

6.2 TEST OBJECTIVES

The quality objectives of testing the Project Tracker were to ensure complete validation of the software requirements:

- Verify that software requirements are complete and accurate.
- Perform detailed test planning.
- Identify testing standards and procedures that will be used on the project.
- Prepare and document test scenarios and test cases.
- Regression testing to validate that unchanged functionality has not been affected by changes.
- Manage defect tracking process.
- Provide test metrics/testing summary reports.

6.3 TESTING GOALS

The goals in testing this application included validating the quality, usability, reliability and performance of the application. Testing was performed from a black-box approach, not based on any knowledge of internal design or code. Testing was done around requirements and functionality.

Another goal was to make the tests repeatable for use in regression testing during the project lifecycle, and for future web application upgrades. A part of the approach in testing was initially perform a ‘Smoke Test’ upon delivery of the application for testing. Smoke Testing is typically an initial testing effort to determine if a new software version is performing well enough to accept it for a major testing effort. For example, if the new software is crashing frequently, or corrupting databases, the software is not in a stable enough condition to warrant further testing in its current state. This testing was performed first. After acceptance of the build delivered for system testing, functions were tested based upon the designated priority (critical, high, medium, low).

6.4 WHAT WERE TESTED

The following features of the website were tested for accuracy

- Home page
- Index page
- Detail page of each Project Tracker
- Loading of each page.

- Login
- Creating, Editing of tasks, minutes, email from Admin, Supervisor, Project Leader Panel.
- Marking System

6.5 TEST EXECUTION

The test execution phase was the process of running test cases against the software build to verify that actual results meet the expected results. Defects discovered during the testing cycle were fixed, the fixed code was incorporated into the application and regression tested.

The following testing phases was completed:

1. Unit testing
2. Functional Testing
3. Regression testing
4. Integration testing
5. Interface testing
6. User acceptance testing
7. Browser Testing

6.6 TEST RESULT

The test result of each unit test and integration test are done while developing the system and is reviewed to identify and remove errors. The

following table consists of the test result of Black box testing which are performed to validate the system with respect to the requirement.

Id	Test	Expected Behavior	Precondition	Actual Behavior	Result
1.	Login	Go to home page	Correct Email/ Password	Go to Home page	pass
2.	Login	Re-Load login page	Incorrect Email/Password	Re-Load login page	pass
3.	Login	Set-Cookie	Remember me is checked	Set-Cookie	pass
4.	Google Login	Go to home page	Registered Google User on Google	Go to home page	pass
5.	Register	Redirect to login page with verification email sent	User details with correct email provided	Redirect to login page with verification email sent	pass
6.	Register	Go to home page	Email verified by user	Go to home page	pass

7.	Logout	Go to Welcome page	User already logged in	Go to Welcome page	pass
8	Add New Project	New Project Created	Admin logged in and members of project registered	New Project Created	pass
9	Show Project Details	Show Project Details, minutes, members, tasks, progress	Minutes added with tasks	Show Project Details, minutes, members, tasks, progress	pass
10	Minute	Add Minute	Project leader logged in	Add Minute	pass
11	tasks	Add/ Delete tasks	Leader logged in	Add/ Delete tasks	pass
12	QR Generate	QR generated	Project Head Logged in and project selected	QR generated	pass
13	QR SCAN	Attendance Done	Supervisor and team	Attendanc e Done	pass

			member scan QR for the project		
14	Notification	Send Email to selected	Select Project	Send Email to selected	pass
15	File upload	Upload ppt, pdf, image, doc	Select Project	Upload ppt, pdf, image, doc	pass
16	Marks	System suggested	Tasks added and attendance	System suggested	pass
17	Marks	Update marks	Supervisor update marks	Update marks	pass

Table 6.1 Test Result

Chapter 7

RESULT AND DISCUSSION

We started to develop the Project Tracker as our final year major project for the Bachelor of Engineering Software Engineering. Within the time frame we were proud to complete this project when we submitted the proposal report for commencing this project.

This project was completed within the given time frame with active involvement and coordination of all the three team members and excellent support from the supervisor.

With the goal, which we started this project, we have come far ahead, now can now hand over our project to our college. All project members, leader, supervisors and admin can use Project Tracker in efficient and effective way.

Chapter 8

FUTURE IMPROVEMENTS

For our next version some of the future improvements of this projects are:

- Fully automatic individual student mark suggestion
- Android and iOS mobile application
- Adding features like documents validation

Chapter 9

CONCLUSION

On completion of this project, a web-based application is made that is able to suggest marks for each member of project based on their active involvement on project completion.

This application reduces the paper work. Details of meetings as well as project progress is obtained. Moreover, users are able to see another projects progress, this creates competitive environment. Thus, projects are likely to be completed within time.

BIBLIOGRAPHY

Apache Friends. (2018). Retrieved from XAMPP Installers and Downloads for Apache Friends: https://www.apachefriends.org/index.html

Atlassian. (2011). Retrieved from Atlassian: https://trello.com/about

Freedcamp., F. (n.d.). Retrieved from Freedcamp - Free Project Management: https://freedcamp.com/

Microsoft. (2018). Retrieved from Visual Studio Code - Code Editing. Redefined: https://code.visualstudio.com/docs

Nils Adermann, J. B. (2012). Retrieved from Composer: https://getcomposer.org/doc/00-intro.md

Oracle. (2018). Retrieved from MySQL: https://www.mysql.com/why-mysql/

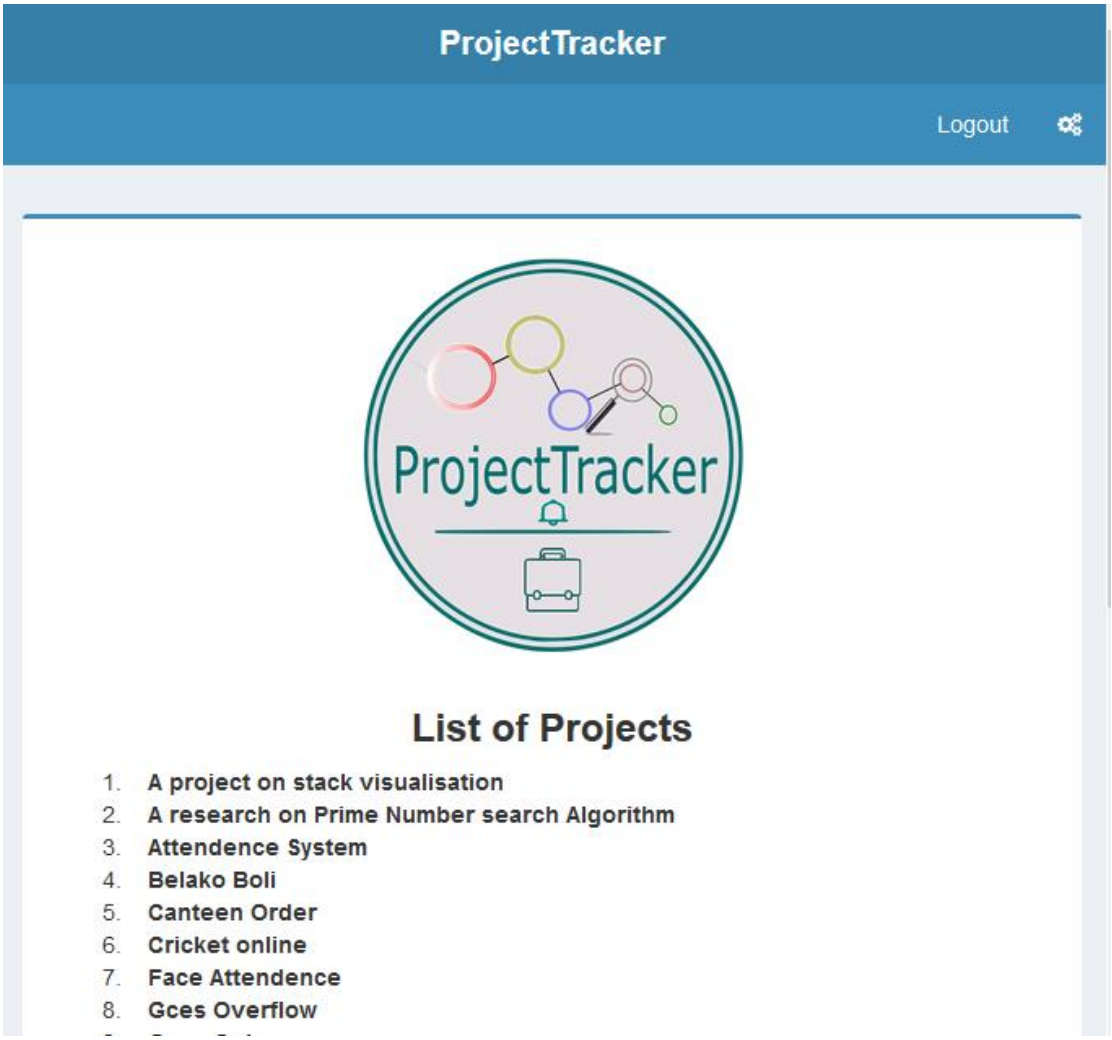
Otto, M. (2018). Retrieved from Bootstrap · The most popular HTML, CSS, and JS library in the world.: https://getbootstrap.com/docs/4.1/getting-started/introduction/

OTWELL, T. (2011). Retrieved from Laravel - The PHP Framework For Web Artisans: https://laravel.com/docs/5.5

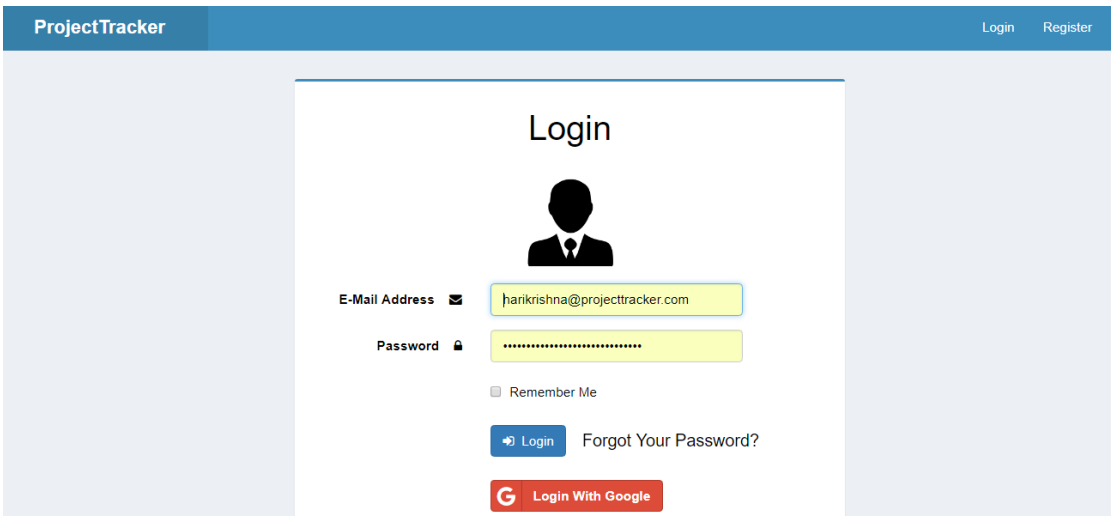
Pluralsight. (2018). Retrieved from Javascript.com: https://www.javascript.com/about

Symphony. (2018). Retrieved from Symfony at a Glance: https://symfony.com/at-a-glance

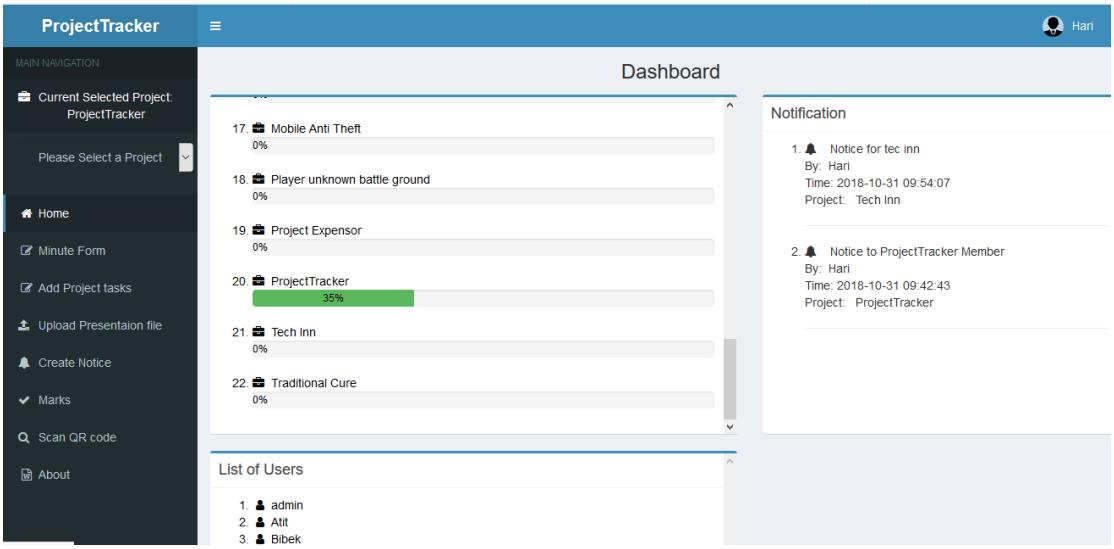
APPENDICES



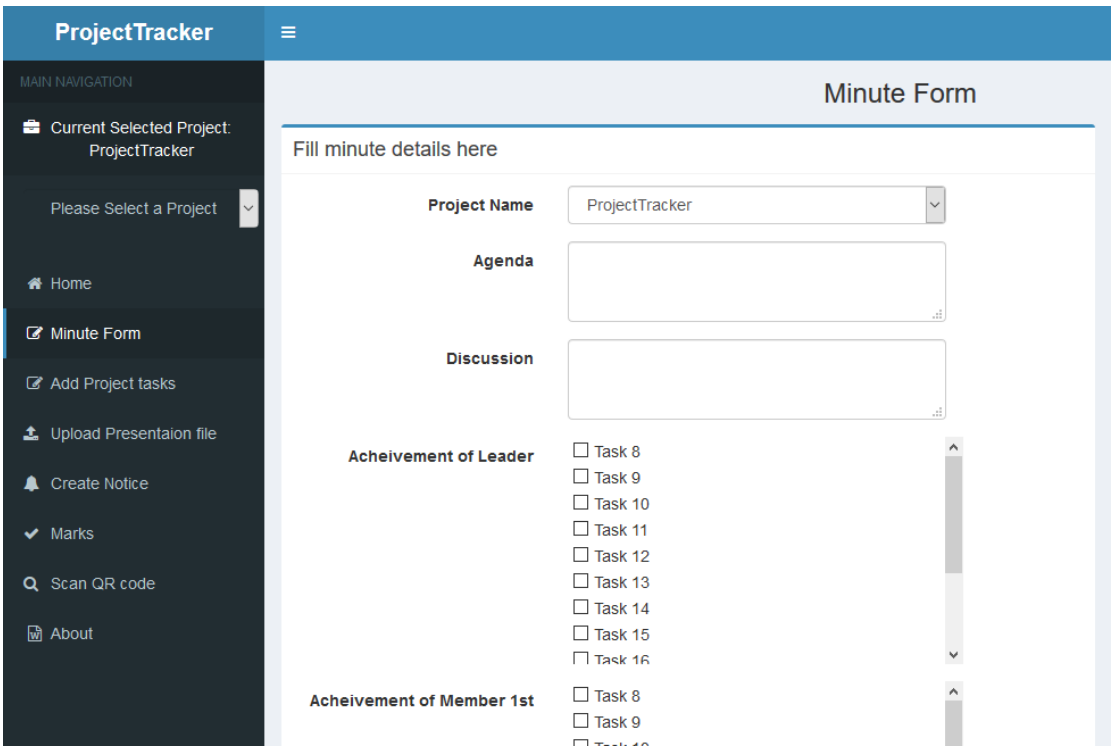
Screenshot 1 Welcome Screen



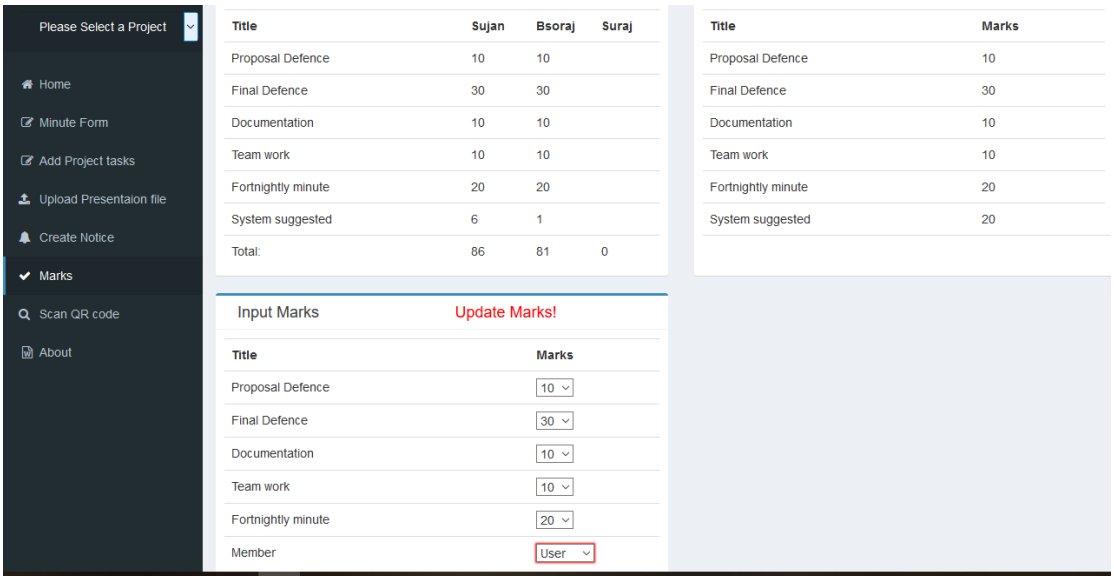
Screenshot 2 Login Screen



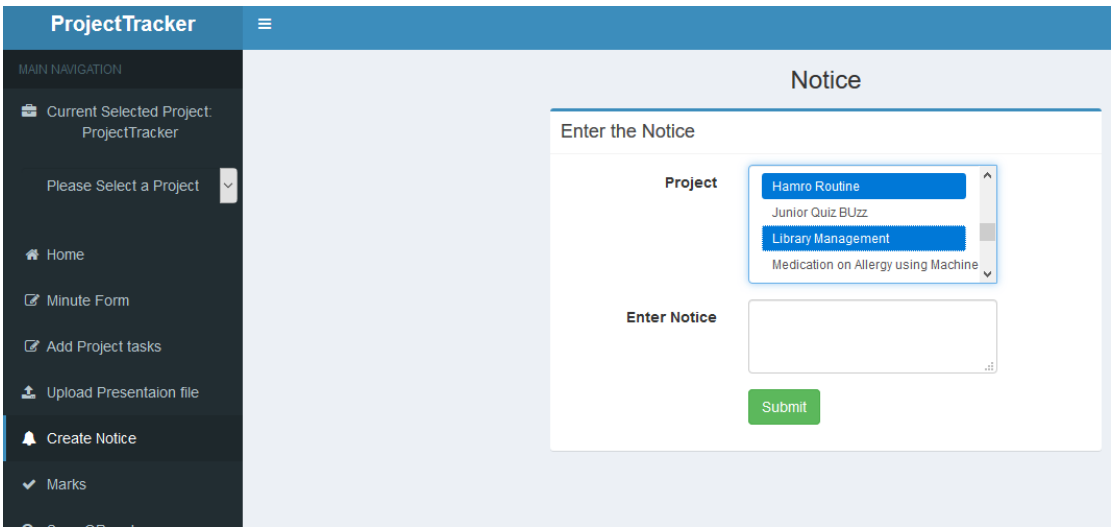
Screenshot 3 Dashboard



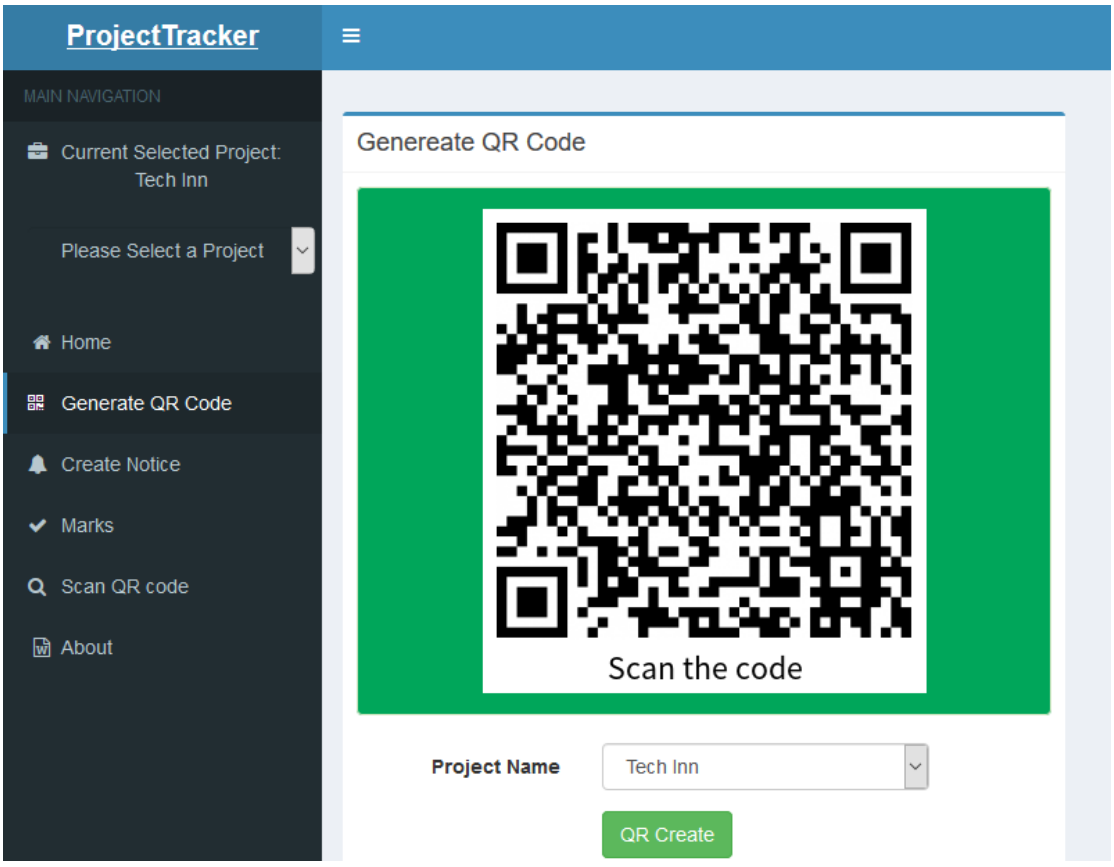
Screenshot 4 Minute Form



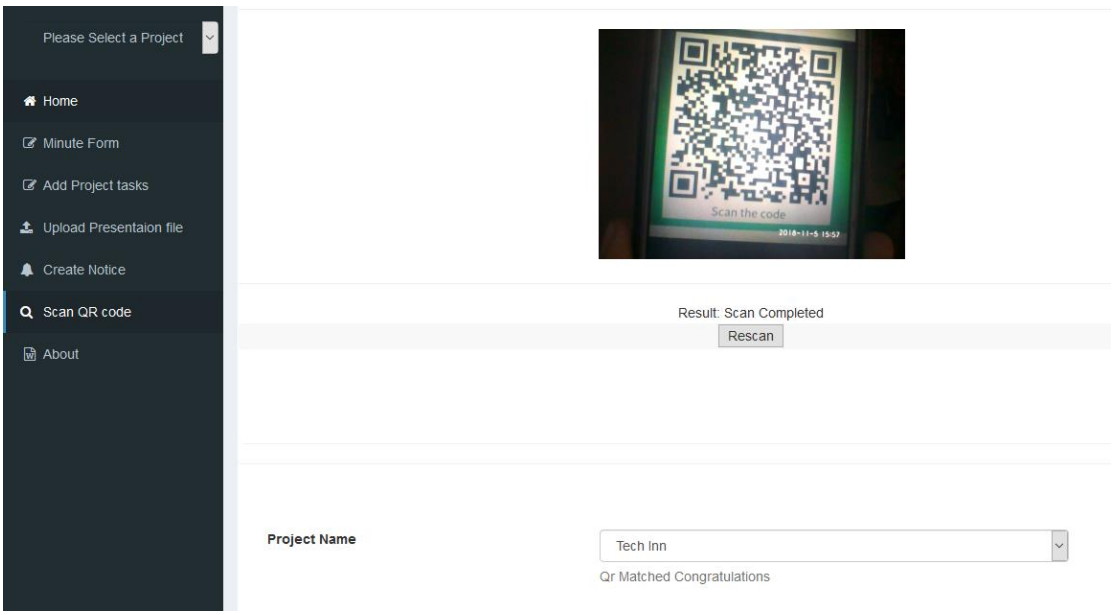
Screenshot 5 Marks Form and view



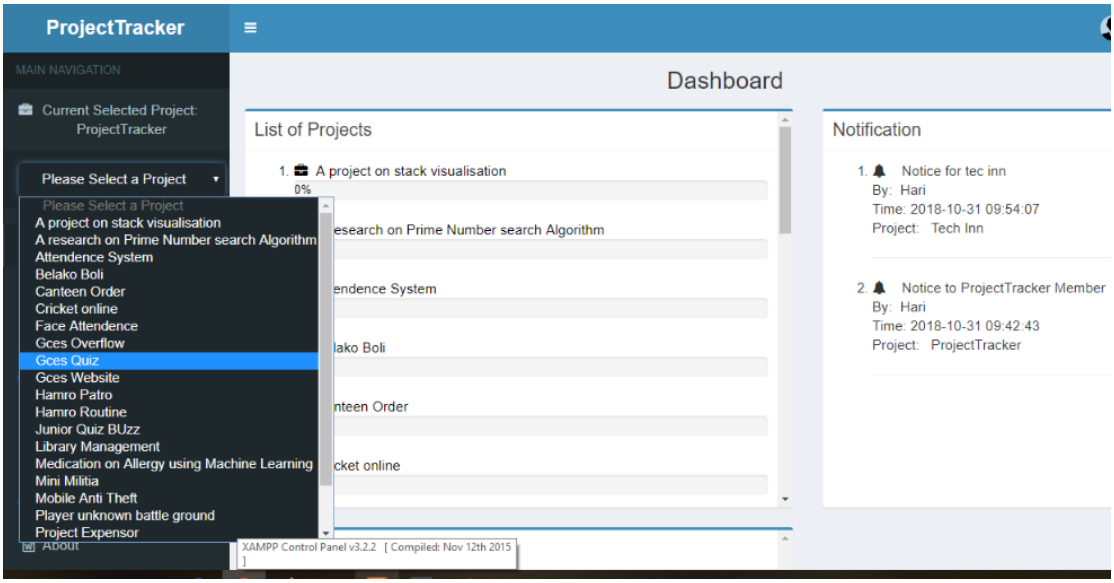
Screenshot 6 Notice Form



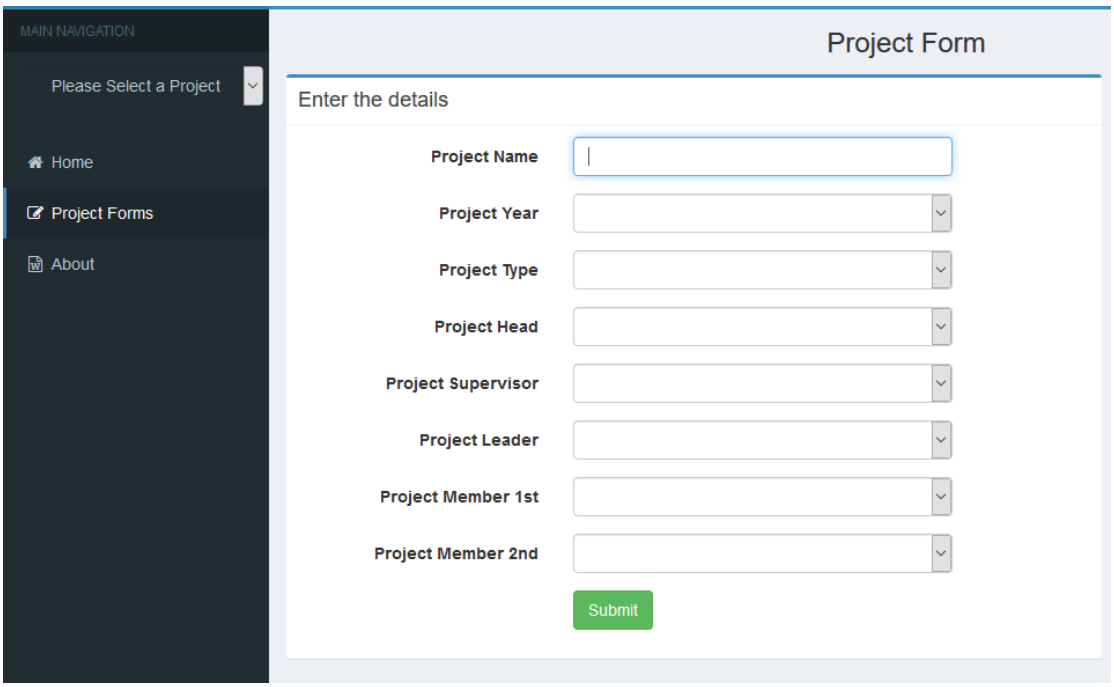
Screenshot 7 QR code Generate



Screenshot 8 QR Scan



Screenshot 9 Project Selection



Screenshot 10 Project Form