

Governors State University OPUS Open Portal to University Scholarship

All Capstone Projects

Student Capstone Projects

Spring 2017

Digitized Engineering Notebook

Syed Jalaluddin
Governors State University

Abdul Khaleeq Mohammed
Governors State University

Azmath Ali Mohammed
Governors State University

Syed A. Samad
Governors State University

Follow this and additional works at: <http://opus.govst.edu/capstones>

 Part of the [Computer Sciences Commons](#)

Recommended Citation

Jalaluddin, Syed; Mohammed, Abdul Khaleeq; Mohammed, Azmath Ali; and Samad, Syed A., "Digitized Engineering Notebook" (2017). *All Capstone Projects*. 298.
<http://opus.govst.edu/capstones/298>

For more information about the academic degree, extended learning, and certificate programs of Governors State University, go to http://www.govst.edu/Academics/Degree_Programs_and_Certifications/

Visit the [Governors State Computer Science Department](#)

This Project Summary is brought to you for free and open access by the Student Capstone Projects at OPUS Open Portal to University Scholarship. It has been accepted for inclusion in All Capstone Projects by an authorized administrator of OPUS Open Portal to University Scholarship. For more information, please contact opus@govst.edu.

ABSTRACT

Digitized Engineering Notebook is a web application that allows students to record their work as they go through the process of their robotics competition. This notebook will portray the same qualities as the current engineering notebook. This digital version of an engineering notebook gives students a platform where they can upload data such as images, texts, videos to a database. Digitized engineering notebook will allow multiple users to log into the system, and work on same project at the same time.

This web application will be developed in an open source environment called XAMPP. It will include multiple users. This Notebook can be categorized into queries which can be converted into PDF. The web application will have login credentials for every user and will also allow name-wise search of any user so their progress can be tracked. The functionalities of this Notebook will include a front page with menu, options to create new projects, addition of new members to the project, table of contents, etc.,

This web application will also contain a calendar so students can mark events. It will have a discussion board where students can interact with each other.

The main goal of this web application is to inspire students to learn STEM engineering process. This Digitized Notebook will encourage knowledge sharing. The contents of the Notebook will be easy to query and share. It will allow students to work as a team remotely. This web application will be kids friendly and will allow students from K-12 to use it.

Table of Content

1	<i>Project Description</i>	1
1.1	Competitive Information	1
1.2	Relationship to Other Applications/Projects	1
1.3	Assumptions and Dependencies	1
1.4	Future Enhancements	1
1.5	Definitions and Acronyms	1
2	<i>Project Technical Description</i>	1
2.1	Application Architecture	2
2.2	Application Information flows	3
2.3	Interactions with other Projects (if Any)	8
2.4	Interactions with other Applications	9
2.5	Capabilities	9
2.6	Risk Assessment and Management	9
3	<i>Project Requirements</i>	9
3.1	Identification of Requirements	9
3.2	Operations, Administration, Maintenance and Provisioning (OAM&P)	12
3.3	Security and Fraud Prevention	12
3.4	Release and Transition Plan	12
4	<i>Project Design Description</i>	13
5	<i>Project Internal/external Interface Impacts and Specification</i>	22
6	<i>Project Design Units Impacts</i>	22
6.1	Functional Area/Design Unit A	22
6.1.1	<i>Functional Overview</i>	22
6.1.2	<i>Impacts</i>	22
6.1.3	<i>Requirements</i>	22
7	<i>Open Issues</i>	23
8	<i>Acknowledgements</i>	23
9	<i>References</i>	23
10	<i>Appendices</i>	23

1 Project Description

The primary purpose of our web application, digitized engineering notebook, is to support the documentation work of engineering project. It will provide the clear, concise, and chronological entries of project with specific dates. The web application allows each member to work and update the project information on engineering notebook. In addition to text, team members can update graphical information. The dynamic web application allows to extract the information updated by each member by the name of member with respect to date and time.

1.1 Competitive Information

Digitized Engineering Notebook will replace regular notebooks where students record their experiments. At the moment, the students use notebooks, tablets etc., to record their work. The idea of digitization of a notebook will make the students' jobs easier. This Digitized Engineering Notebook has the potential to be the first of its kind in the market as it provides students/members with same features as a regular notebook.

1.2 Relationship to Other Applications/Projects

This web application relates to Plumbago. Plumbago paper app lets the users sketch and Handwrite notes. But, in Digitized Engineering Notebook, not only you can add text but also you can add media to logs.

1.3 Assumptions and Dependencies

- The web application can handle huge traffic.
- The web application can handle thousands of projects.
- The user must have active internet.
- The user should have an operating system which supports browsers with rich HTML code.

1.4 Future Enhancements

This is currently the only phase of this project. However, we plan to add the feature of version control where the captain will be able to retrieve all the logs entered by members even if they were removed. The first log entered is considered as version 1. Similarly, the edited version of that log would be called version 2.

1.5 Definitions and Acronyms

The acronym used in this project is DEN which is the SQL file of the database. Definitions used in this project are Web Application, Centralized database

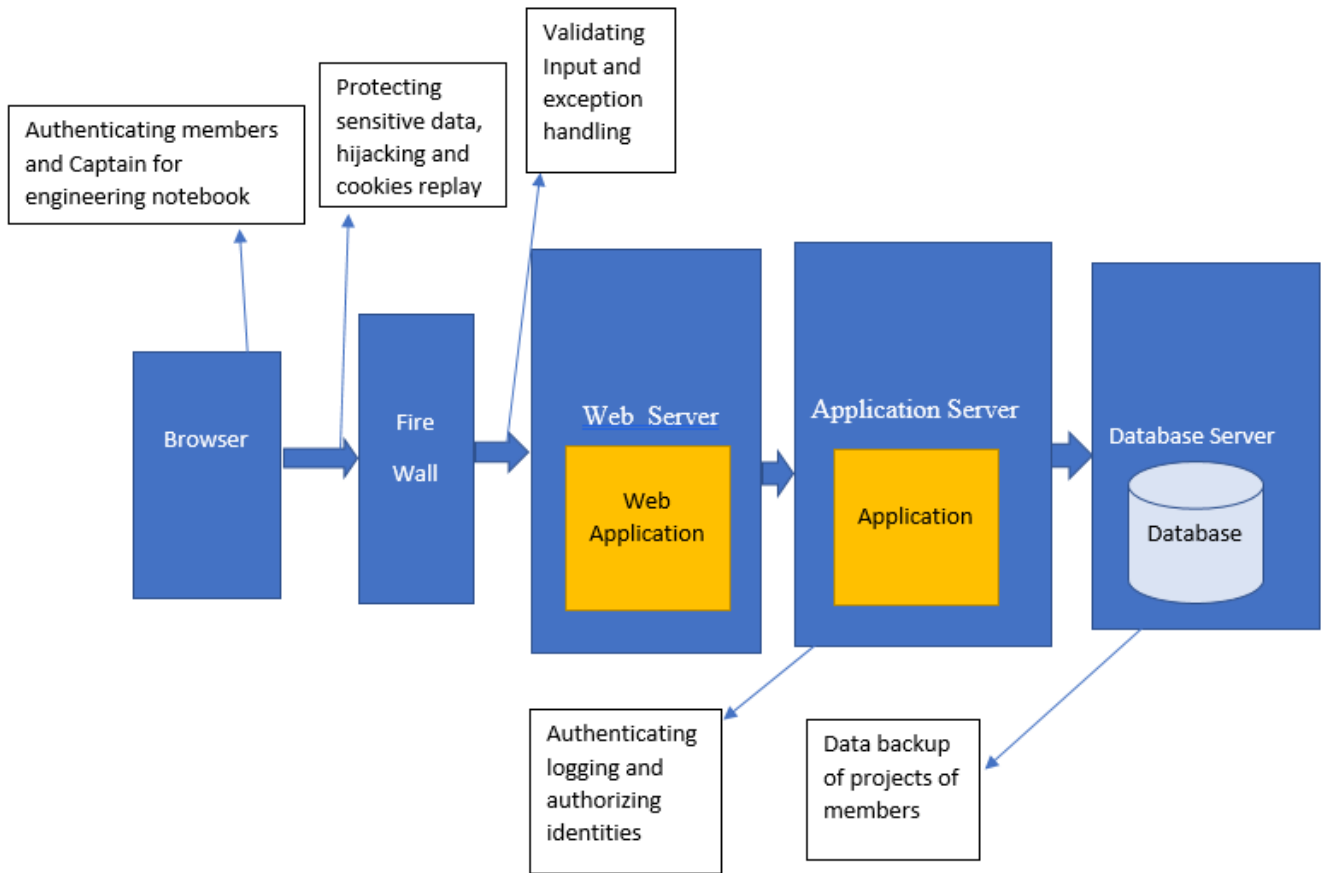
Web Application interacts with the user on a server.

Centralized Database is stored and maintained on a specific location this could be a central computer or a database system.

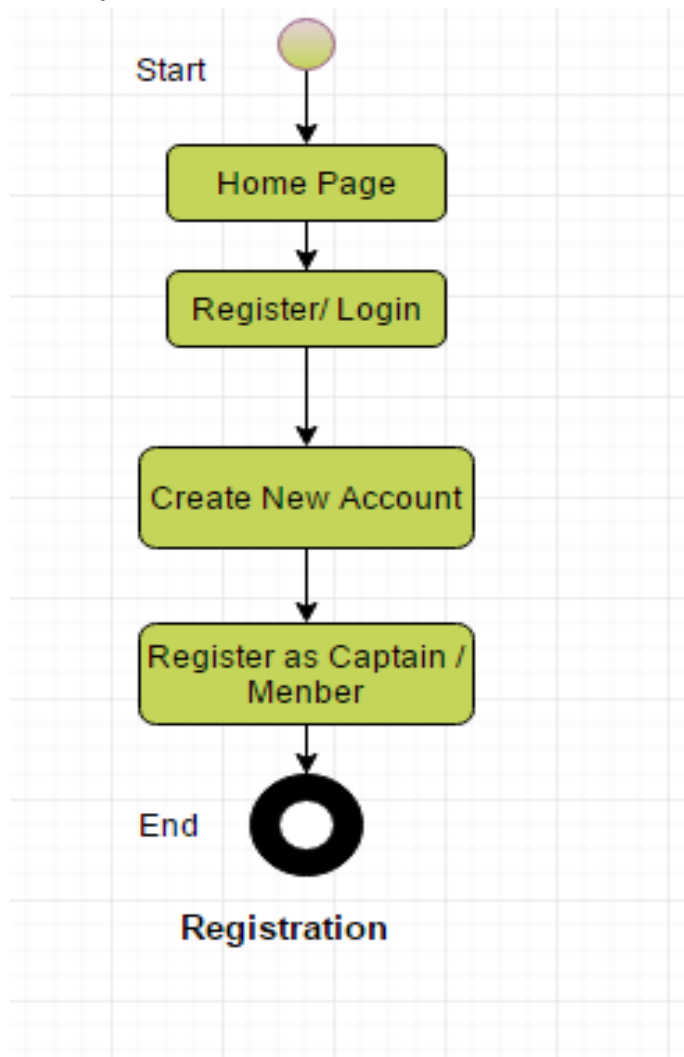
2 Project Technical Description

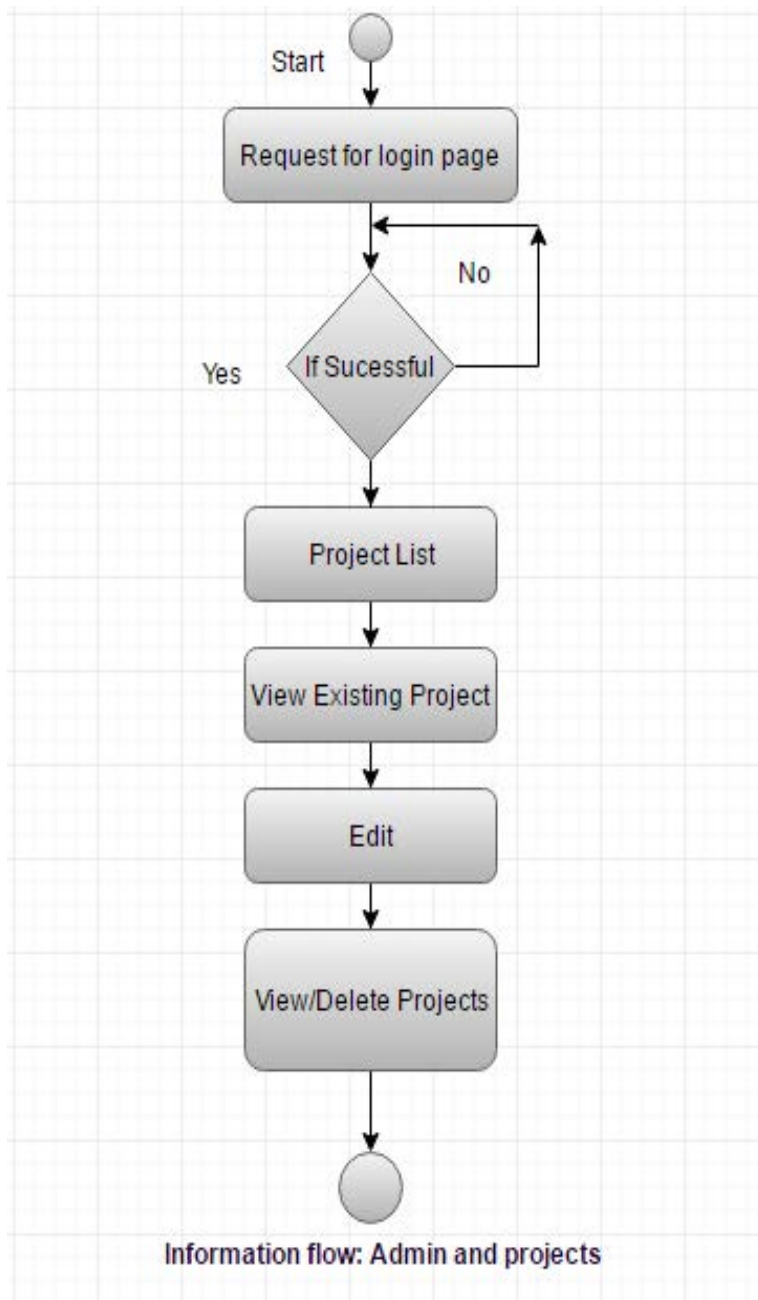
PHP is the language used to create the application. This web application was made using an open source environment called XAMPP. XAMPP allows the use of Apache server and SQL. With Apache and SQL running in the system, the host computer can act as the server. The database was made using SQL and can be accessed by clicking on PHPMyAdmin on the XAMPP application. The database can handle many users by allowing any number of members on a team and can also allow any number of projects to run at the same time.

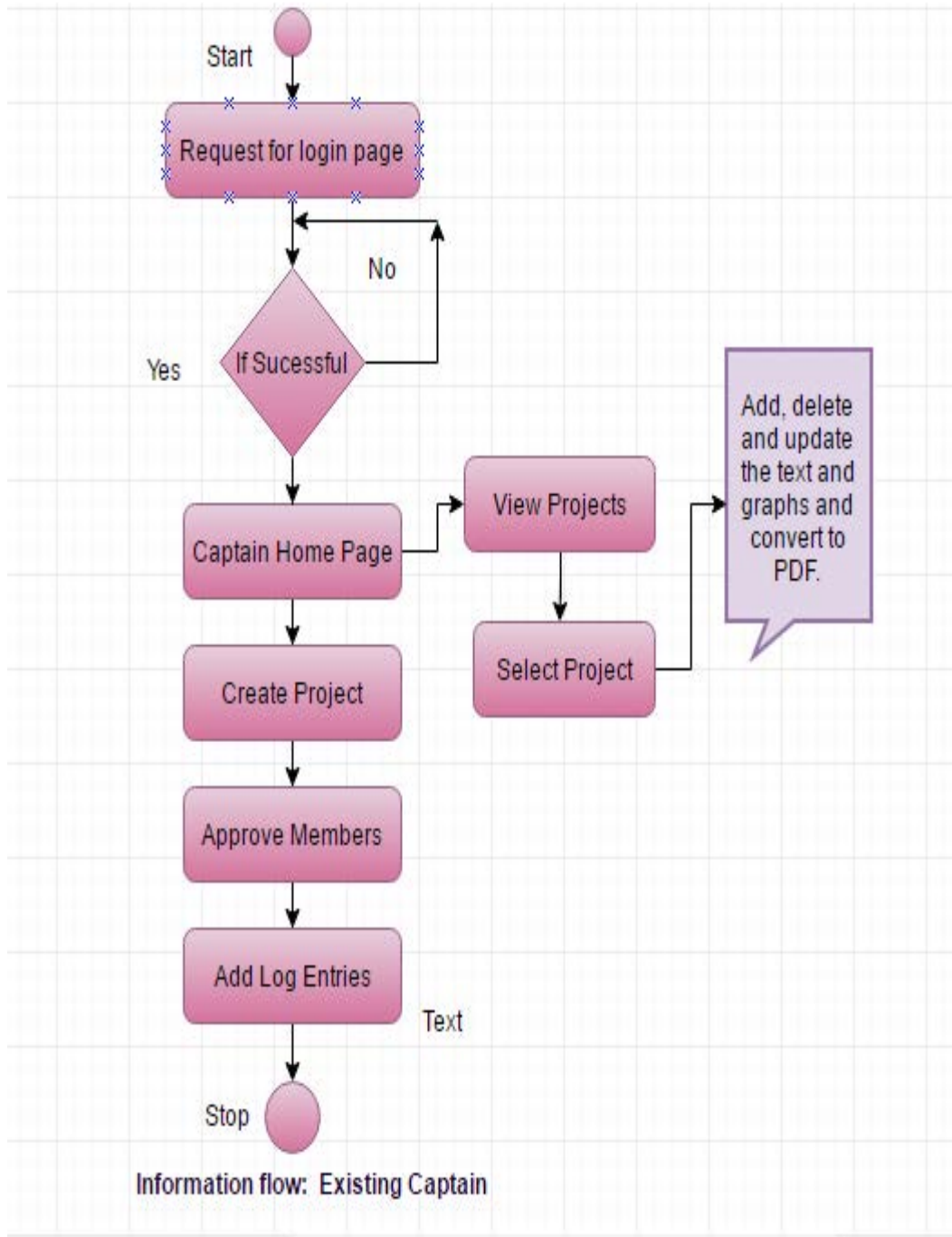
2.1 Application Architecture

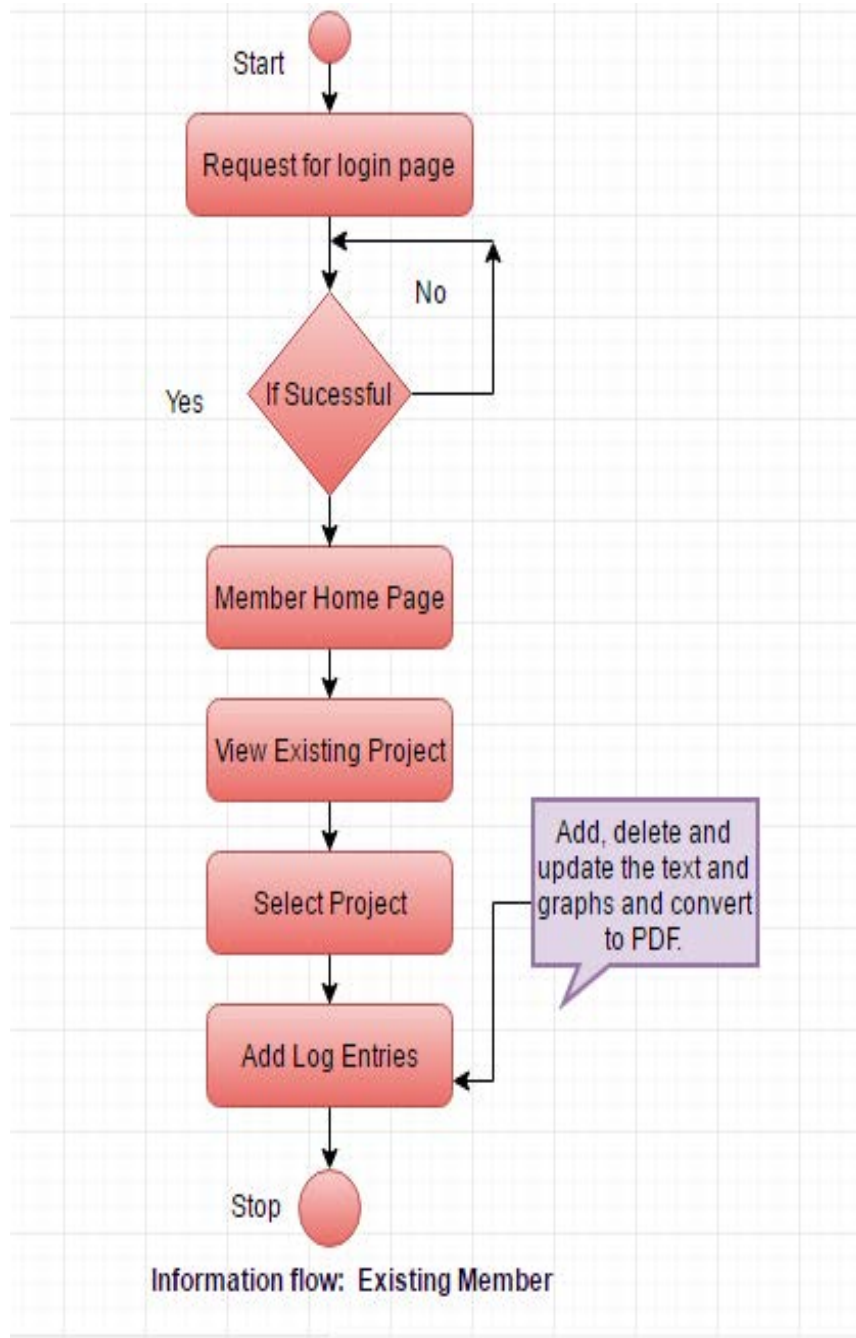


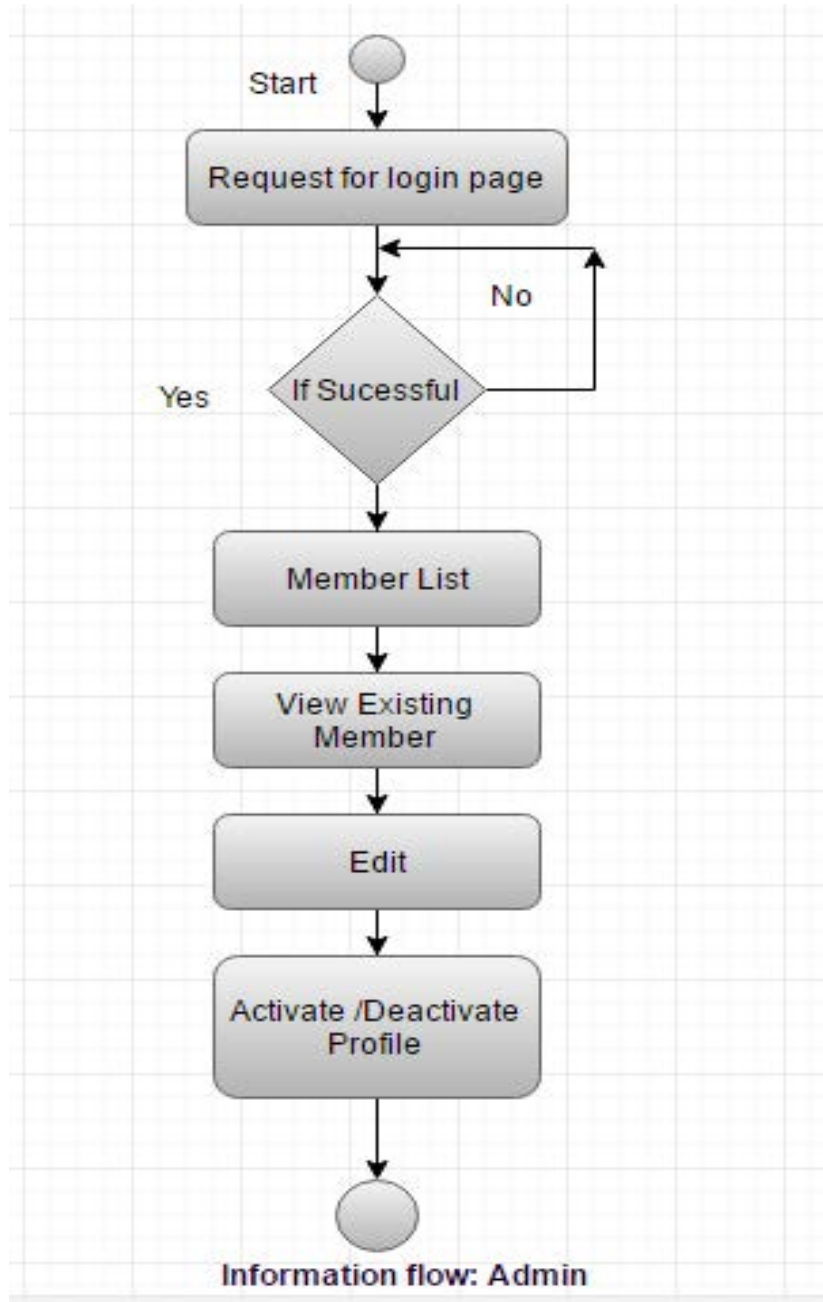
2.2 Application Information flows

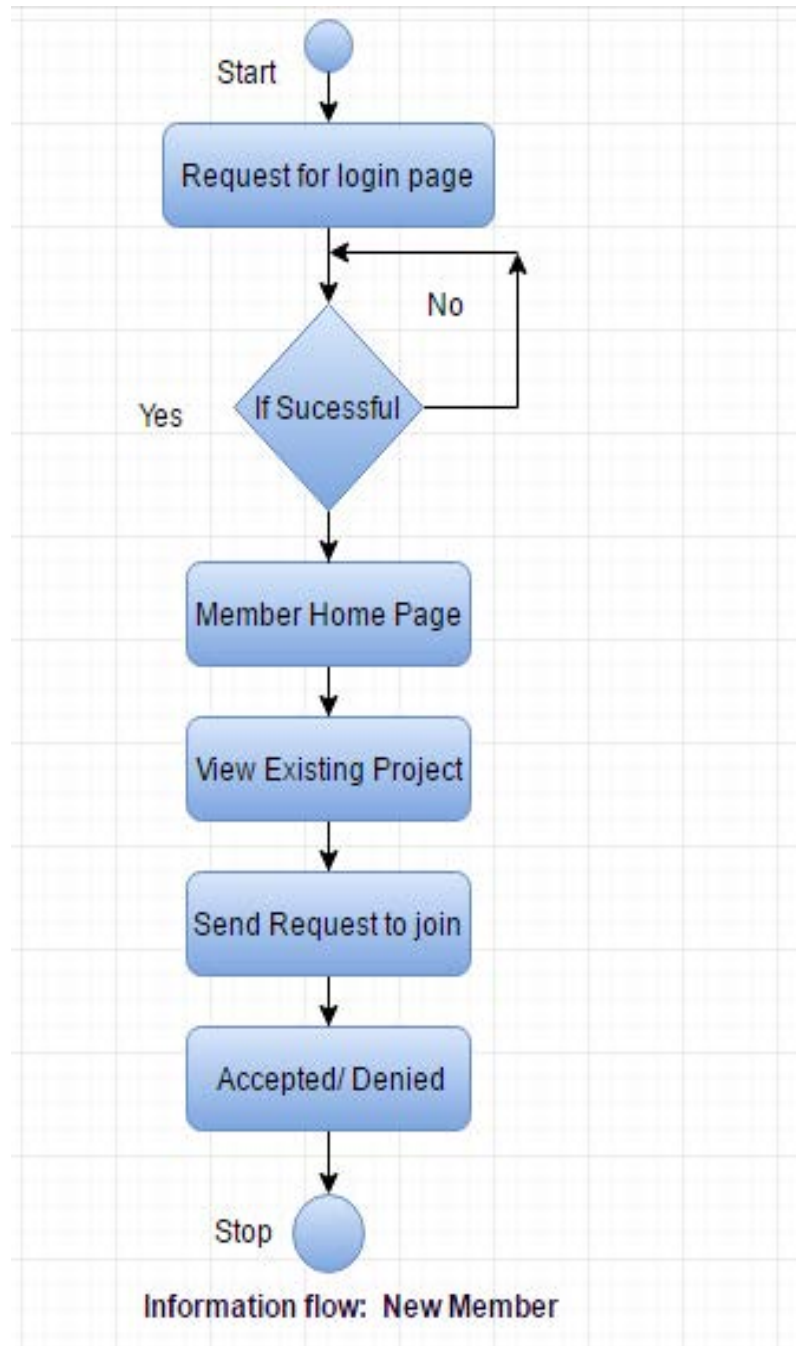












2.3 Interactions with other Projects (if Any)

None.

2.4 *Interactions with other Applications*

None.

2.5 *Capabilities*

The capabilities of this Web application include functionalities of admin, captain and member. The admin in this web application will be able to login, and he will approve the captain and view the members, projects and teams.

The captain will create the project and he will approve the members who request to be on the team. The captain basically has the same functionalities as the member except he/she can edit the project.

The members will be able to sign up and add log entries and such as text, images, and graphs. Members can also view project list and the details of the projects they are a member of. They can also convert the project to PDF.

2.6 *Risk Assessment and Management*

This Project contains the sensitive information. Since the sensitive information is vulnerable, end to end encryption is implemented to secure the network.

Another risk is the loss of valuable information or data and to avoid this risk, we are maintaining regular backups of data. If there is a huge flow of traffic there is a chance of server going down. To enhance our server capabilities, we will be using the high-end server that can withstand higher overheads of traffic.

If a user opens our web application in a tablet or mobile phone there is a chance of bad screen resolution. So, to increase the better user experience in mobile devices, we will be using the CSS and Bootstrap. HTTP protocol is implemented to avoid the website being compromised by hackers.

3 *Project Requirements*

3.1 *Identification of Requirements*

<Web-Experts-786-Member-Capability-“1”>

The project must allow members to add text to the logs.

Implementation: Mandatory.

<Web-Experts-786-Member-Capability-“2”>

The project must allow members to request team captains to be on the team.

Implementation: Mandatory.

<Web-Experts-786-Member-Capability-“3”>

The project must allow members to add images to the logs.

Implementation: Mandatory.

<Web-Experts-786-Member-Capability-“4”>

The project must allow members to search for projects.

Implementation: Mandatory.

<Web-Experts-786-Member-Capability-“5”>

The project must allow members to convert logs into PDF.

Implementation: Mandatory.

<Web-Experts-786-Member-Capability-“6”>

The project must allow new members to sign up.

Implementation: Mandatory.

<Web-Experts-786-Member-Capability-“7”>

The project must allow members to specifically search for team-member’s logs.

Implementation: Mandatory.

<Web-Experts-786-Member-Capability-“8”>

The project must allow members to communicate with team-members.

Implementation: Mandatory.

<Web-Experts-786-Member-Capability-“9”>

The project must prohibit members from editing the logs.

Implementation: Mandatory.

<Web-Experts-786-Validation-Control-“10”>

The project must validate user credentials using the database.

Implementation: Mandatory.

<Web-Experts-786-Captain-Capability-“11”>

The project must allow Captains to create projects.

Implementation: Mandatory.

<Web-Experts-786-Captain-Capability-“12”>

The project must allow Captains to approve members to be on the team.

Implementation: Mandatory.

<Web-Experts-786-Captain-Capability-“13”>

The project must allow Captains to add text to the logs.

Implementation: Mandatory.

<Web-Experts-786-Captain-Capability-“14”>

The project must allow Captains to add images to the logs.

Implementation: Mandatory.

<Web-Experts-786-Captain-Capability-“15”>

The project must allow Captains to edit the logs.

Implementation: Mandatory.

<Web-Experts-786-Captain-Capability-“16”>

The project must allow Captains to be approved by the admin.

Implementation: Mandatory.

<Web-Experts-786-Captain-Capability-“17”>

The project must allow Captains to view the logs.

Implementation: Mandatory.

<Web-Experts-786-Captain-Capability-“18”>

The project must allow Captains to convert the logs into PDF.

Implementation: Mandatory.

<Web-Experts-786-Admin-Capability-“19”>

The project must allow admin to view projects.

Implementation: Mandatory.

<Web-Experts-786-Admin-Capability-“20”>

The project must allow admin to view members.

Implementation: Mandatory.

<Web-Experts-786-Admin-Capability-“21”>

The project must allow admin to view teams.

Implementation: Mandatory.

<Web-Experts-786-Admin-Capability-“22”>

The project must allow admin to change password.

Implementation: Mandatory.

<Web-Experts-786-Admin-Capability-“23”>

The project must allow admin to log out.

Implementation: Mandatory.

<Web-Experts-786-Database-Capability-“24”>

The project must allow unlimited members to sign up.

Implementation: Mandatory.

<Web-Experts-786-Database-Capability-“25”>

The project must allow members to create their own projects.

Implementation: Mandatory.

<Web-Experts-786-Database-Capability-“26”>

The project must allow unlimited logs to be stored in the database.

Implementation: Mandatory.

<Web-Experts-786-Database-Capability-“27”>

The project must allow images to be stored on the database.

Implementation: Mandatory.

<Web-Experts-786-Database-Capability-“28”>

The project must allow backups to be made.

Implementation: Optional.

<Web-Experts-786-Database-Capability-“29”>

The project must be able to get updated.

Implementation: Mandatory.

<Web-Experts-786-Design-Capability-“30”>

The project must allow changes to be made to the design if need may be.

Implementation: Mandatory.

3.2 *Operations, Administration, Maintenance, and Provisioning (OAM&P)*

Operations: The customer is provided with the administrative privileges so they can regularly maintain the usage and overhead of traffic. The website will be deployed on GoDaddy.com using the steps mentioned in section 3.4. The administrator’s operations are basic operations that are necessary for the security of the web application.

Administration: The customer is advised to perform regular backups to achieve data integrity. Also, the user/admin of this application needs to change their password every two weeks to ensure safety and security of the application.

Maintenance: The customer is advised to maintain high-end firewalls to protect their systems from being compromised. Using genuine operating systems is advised for proper functioning of this application. Anti-virus applications are recommended to protect the server from being compromised.

Provisioning: The customer is advised to regularly check operations of the web application and report any issues to Web-Experts. The customer is advised to report any new changes to be made and wait for Web-Experts to deliver the new provisions.

3.3 *Security and Fraud Prevention*

We are protecting the security of this web application by keeping the software up to date. The passwords we use in the admin area in this web application and the server are very strong. With the use of validators, we are protecting the web application from fraud.

HTTP is used to provide security over the internet. If the user uploads any unethical image, the captain will remove the image and deactivate the user.

The admin has control over members and the projects. The administrator can activate and deactivate members if need may be. The admin is the one who approves the captain. If any member were to post inappropriate text or media using this platform, the admin has the authority to take it down and deactivate the member.

The captain has control over all members in his/her project. All the members need the captain’s approval to be on the team. If any member were to make a mistake in their logs or post inappropriate text/image, the captain has control to edit or remove the posts made by the members. Also, the captain holds the right to deactivate a member if need may be.

3.4 *Release and Transition Plan*

These are the steps involved in the release and transition of our engineering notebook web application. We are planning to host the web application on bluehost.com. The web application will be released on 05/24/2017.

We registered a domain name and created the user account on bluehost.com. Next step was to get hosting services. We paid for the hosting services and we were ready to go. We used File Transfer Protocol(FTP) to upload the files on bluehost.com. There are generally four things needed to upload files on the web hosting service. They are domain name, username, password, and port 21.

Below are the steps we followed to deploy the web application from our localhost to Bluehost using FTP. We logged in using our FTP account provided by Bluehost. The domain tab on top left corner allows you to setup the domain name for your account. After clicking on that. we placed the domain name in the “FTP address” field. This way, the FTP protocol was setup for the domain name we registered for our web application.

The next step was to create a database. We gave the database the same name as in our localhost and assigned that database to our Bluehost account. Then we clicked on the file manager at the top of the screen and uploaded the PHP files from our computer to our FTP account.

Now that we have a domain name, created a database, and transferred the PHP files from our computer, the next step was to upload the database file. This was done by copying the SQL file from PHPMyAdmin on the XAMPP application and pasting it in the same folder for the database that we just created.

We clicked on go button and our web application went live in about 15 minutes.

4 Project Design Description

This web application helps digitize engineering notebook. There will be three users in this web application. Admin, captain, and member. The admin is responsible for the performance security and efficiency of the website. Admin will approve the captain. Admin can view and delete the list of projects and team members.

To register as a captain the user creates account by providing email, personal information and selects the role. Once the captain is approved by the admin, the captain can create projects and approve members to be on the team. The captain can view and edit the logs. Captain will make sure that the articles, language, and images reflect a positive representation of STEM engineering process. Captains can also convert projects into PDF.

The third user will be the member. To log in as a member, member should provide email, first name and last name and select the role type. They can search the projects and request to join them. The members can view the projects they are in. After captain's approval, a member can start adding logs to the project. The member can convert the projects into PDF.

Below are the screenshots from the web application's design. The images are mostly self-explanatory.

Figure 1: Homepage showing login and registration buttons on top right corner of the screen. The screen shows students working on a robotics project. The about us and contact us tabs contain information about the web application and how digitized engineering notebook helps students participate in and learn STEM process.

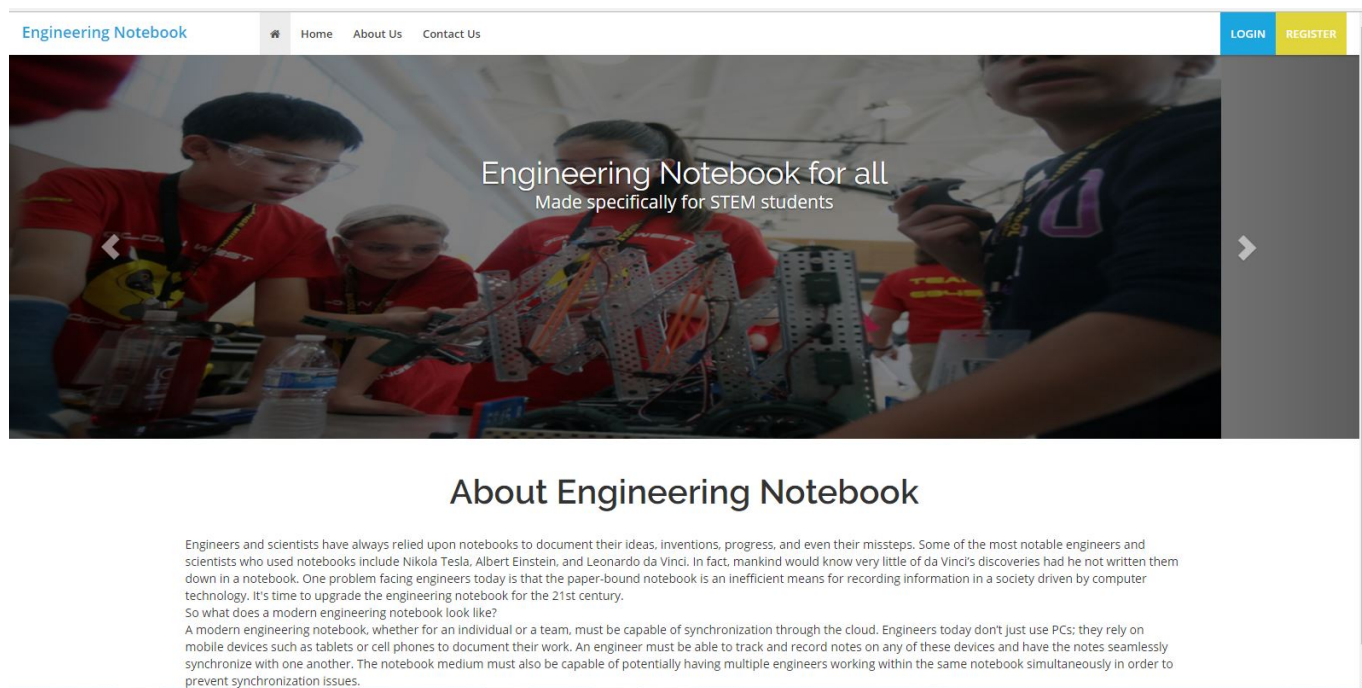


Figure 1 Homepage

Figure 2: Registration page. Allows members to select the role as they register. The role types are Captain and Member. If someone decides to register as the captain, they would need the admin's approval until they can log in and start working as a captain. Upon registration, a member should get to see all the existing projects in the system.

Engineering Notebook

Home About Us Contact Us

LOGIN REGISTER

Create New Account

Account Information

Email *

Password *

Personal Information

First Name *

Last Name *

Role Type *

Sign Up

Figure 2 Registration Page

Figure 3: The login page consists of two fields that are used to authenticate a user. If the user does not already exist, they can click on the sign up now button to register. If the log in details do not match, an error message will be displayed indicating the user does not exist.

Engineering Notebook

Home About Us Contact Us

LOGIN REGISTER

Existing User

Email *

Password *

Sign In

Not a member yet? Click Below to Sign Up

Sign Up Now

Figure 3 Login Page

Figure 4: The admin dashboard consists of multiple buttons. The most useful of them being the members list button and the projects list button. The admin has control over all members and projects. Meaning, he/she can activate or deactivate members or delete logs if the need may be.

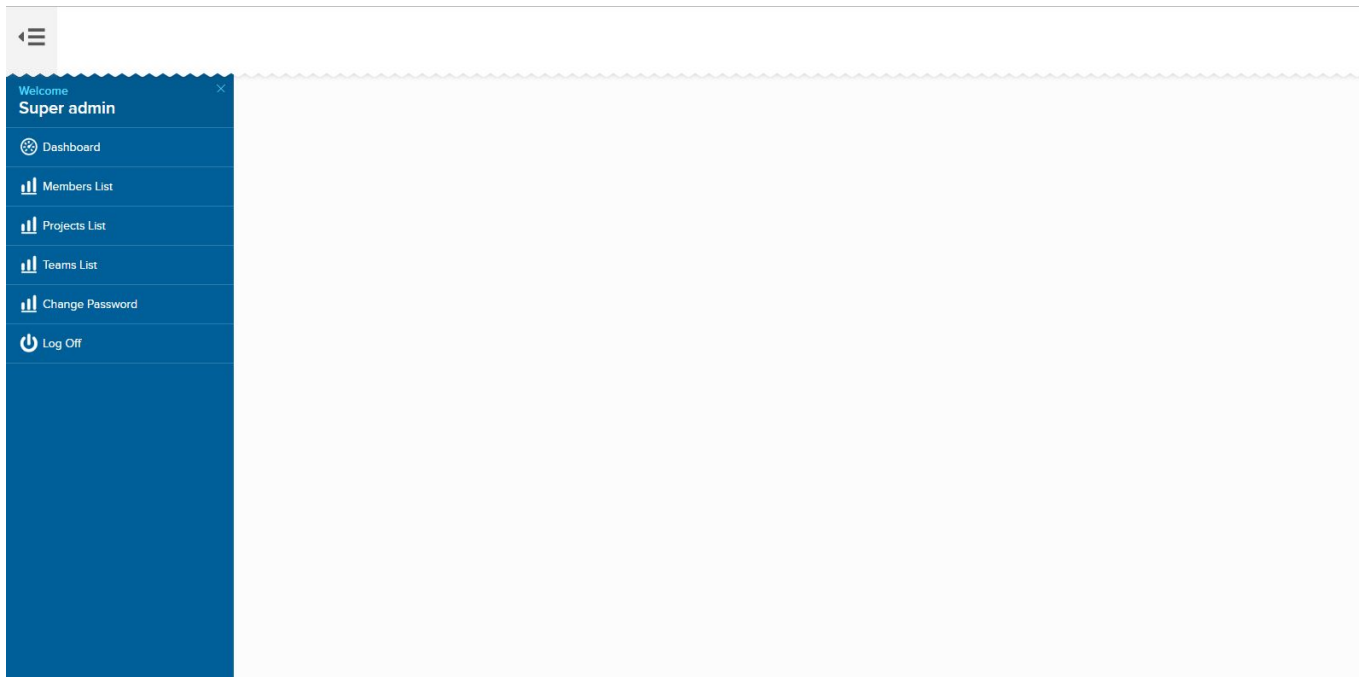


Figure 4 Admin Dashboard

Figure 5: Admin; members list. Admin has control over members. He can activate captains and deactivate members if the need may be. The administrator has access to the database and should be responsible for the operation and maintenance of the web application.

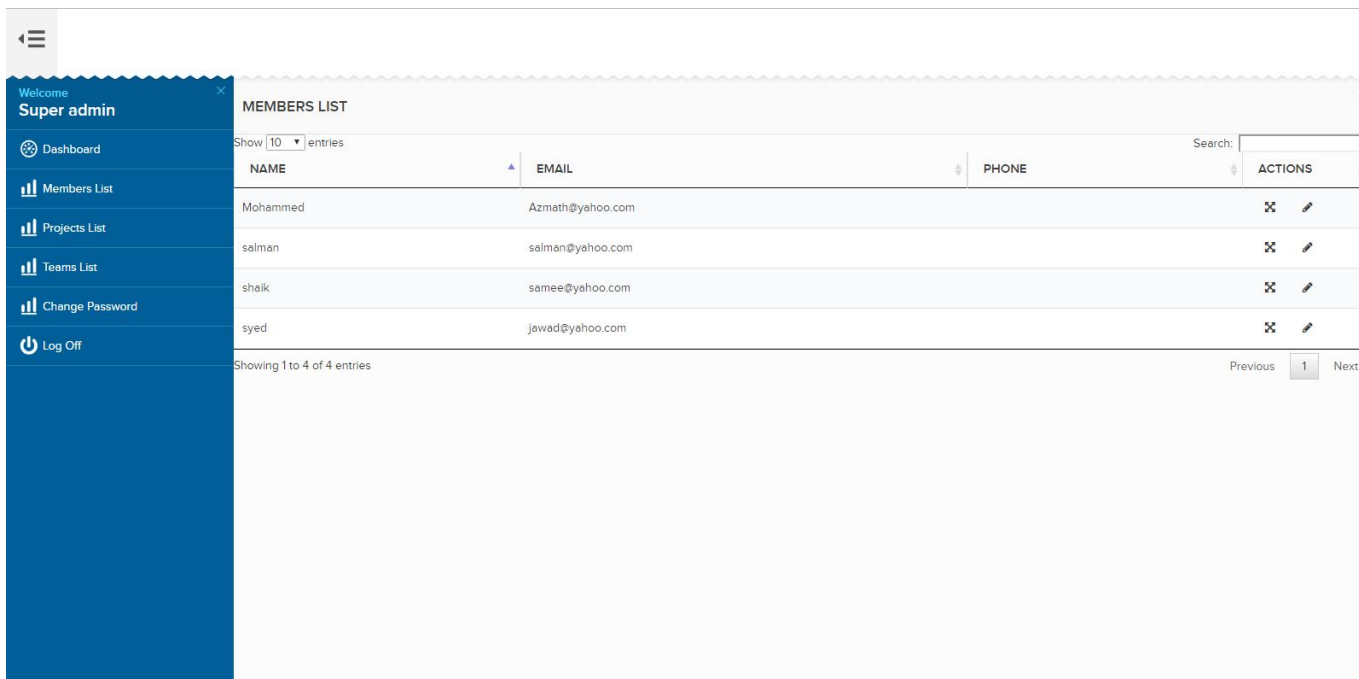


Figure 5 Admin Members List

Figure 6: Admin; Projects list. Shows the existing projects in the system. The admin can delete the projects upon request by captains or by actions of team members.

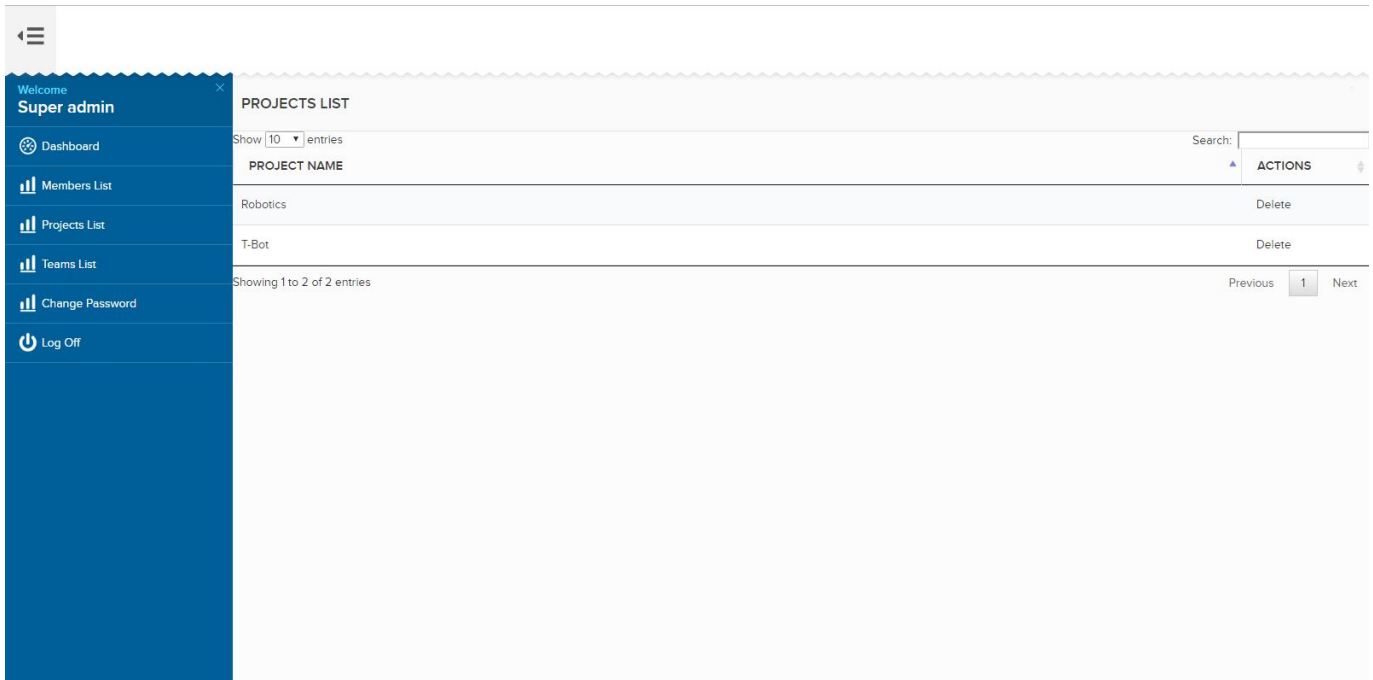


Figure 6 Admin Projects List

Figure 7: Captain; Create Project tab. Allows the Captain to create projects. This page allows the captain to give the project a name. The project description field is a good feature as it allows an in-depth explanation of the project. Also, this page lets the captain add the date as to when the project was created.

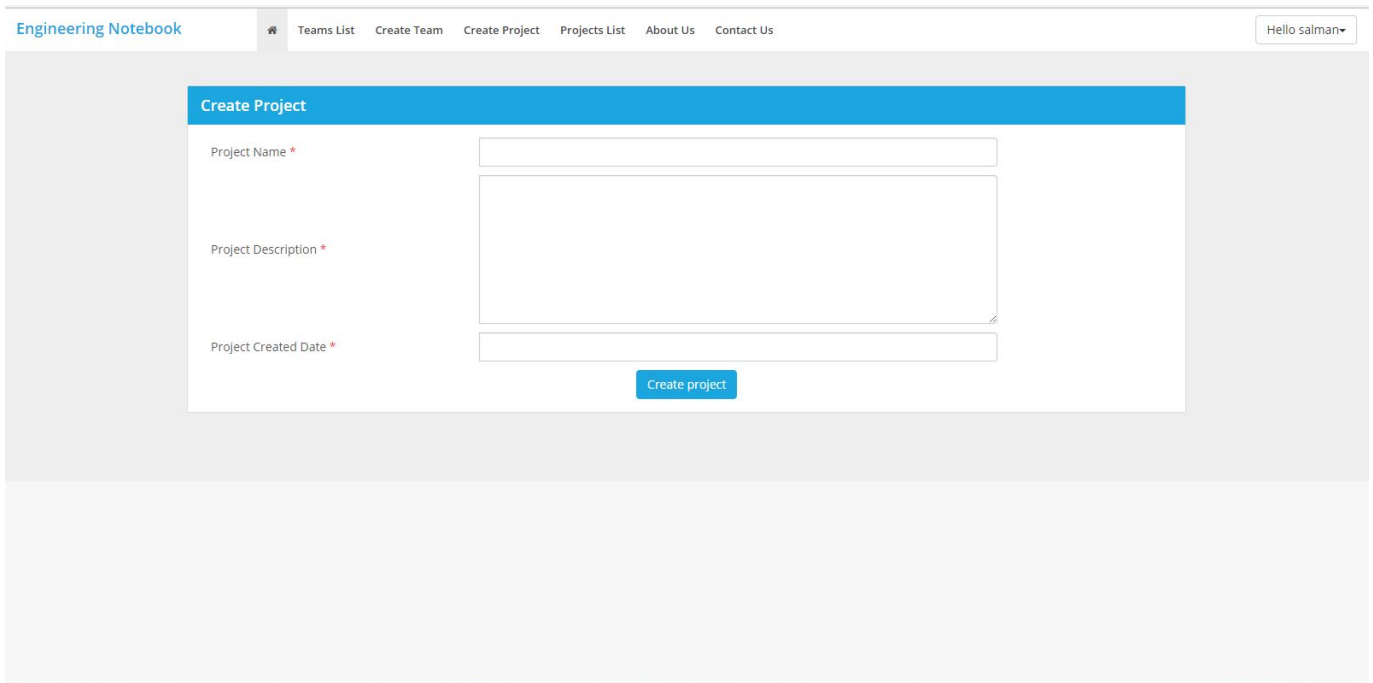


Figure 7 Captain Create Project

Figure 8: Captain; Projects list. Shows the captain the projects he is a captain/member of. The action buttons allow the captain to add log entries to the project, view project requests from members, view the existing logs and edit them.

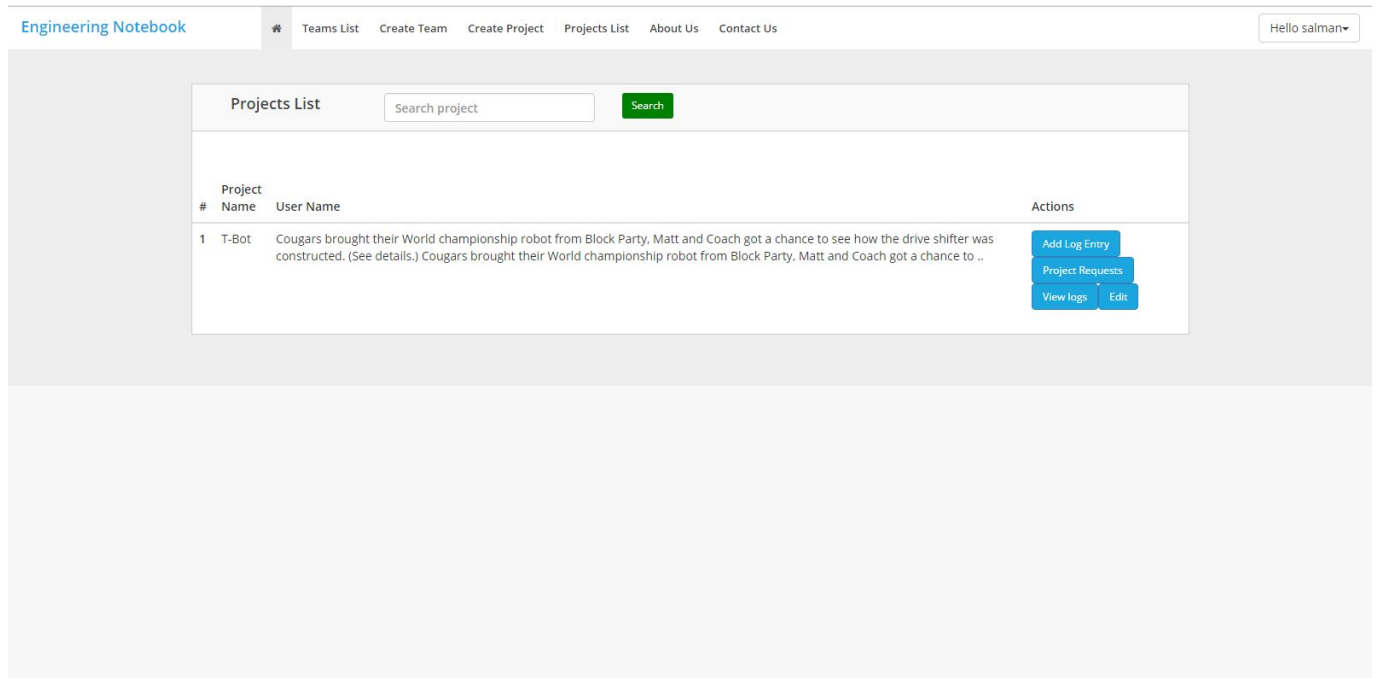


Figure 8 Captain Projects List

Figure 9: Captain; Project Requests page. Shows some information about the members who want to be a part of the project. The fields show the name of the project a member is requesting to join, the name of the member, the member's email address and their status. With the member's email address being visible, the captain can get to know the member by contacting them through email and see if they are good enough for the team.

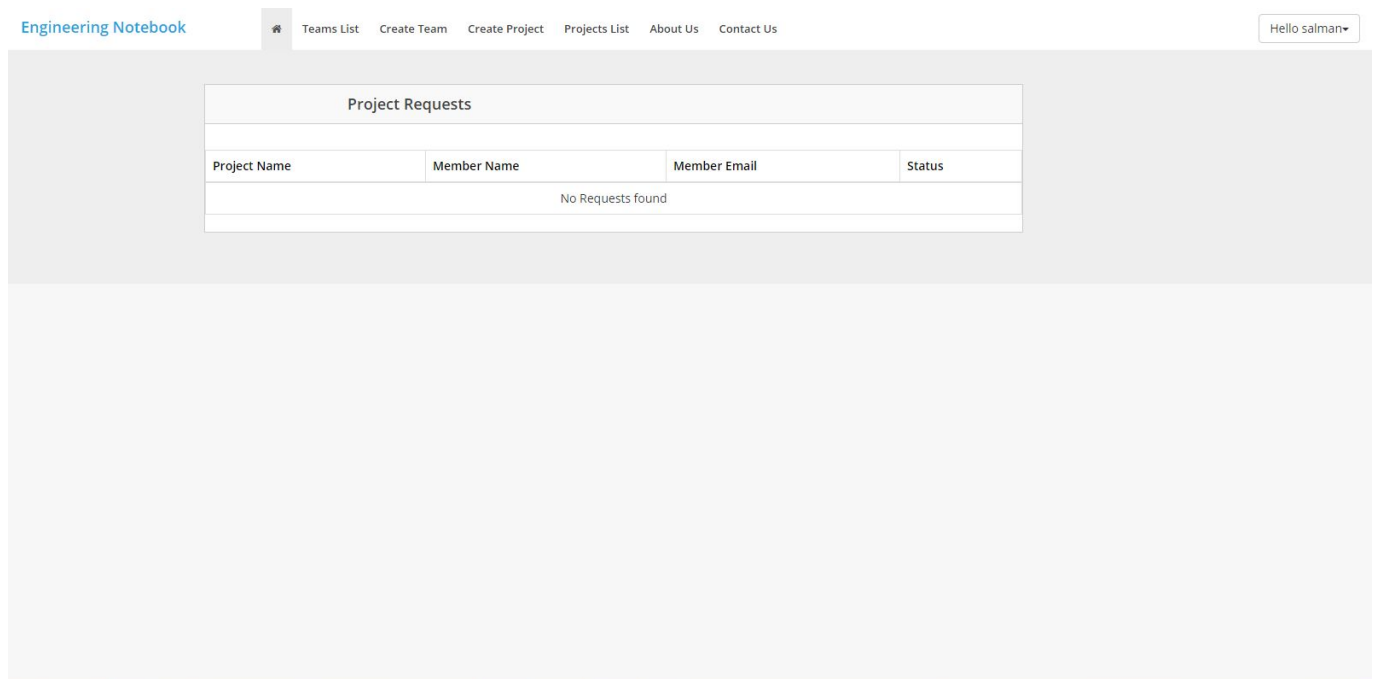


Figure 9 Project Requests

Figure 10: Captain; View Logs page. Shows the logs made by the members in the project. In the figure, T-Bot indicates the name of the project. The page also shows the duration of time it needed to complete it. The most important feature of this page is attendance. It shows the names of the members or special guests that helped complete the project.

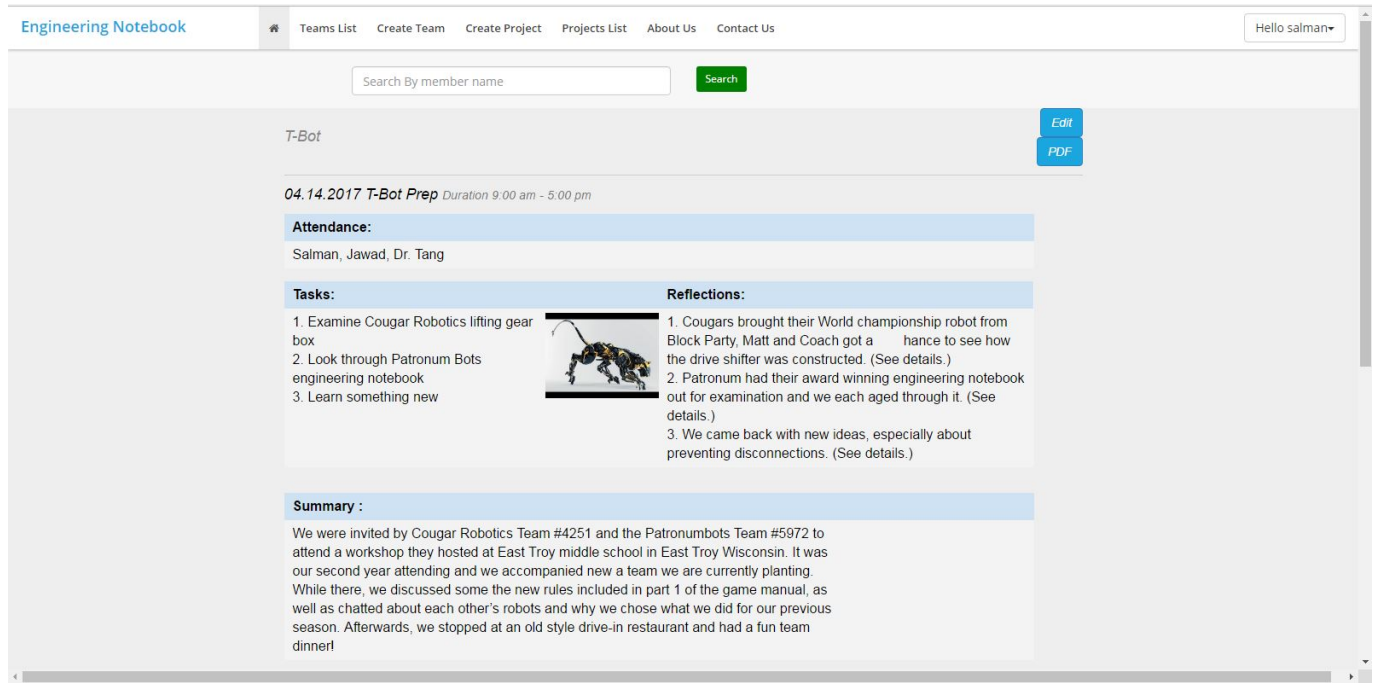


Figure 10 Captain View Logs

Figure 11: Captain; Edit Page allows the captain to remove existing text or replace it with new text. This feature is only given to captain so he/she can correct potential errors made by team members. This also happens to be a good security feature incase a log needs to be edited or removed.

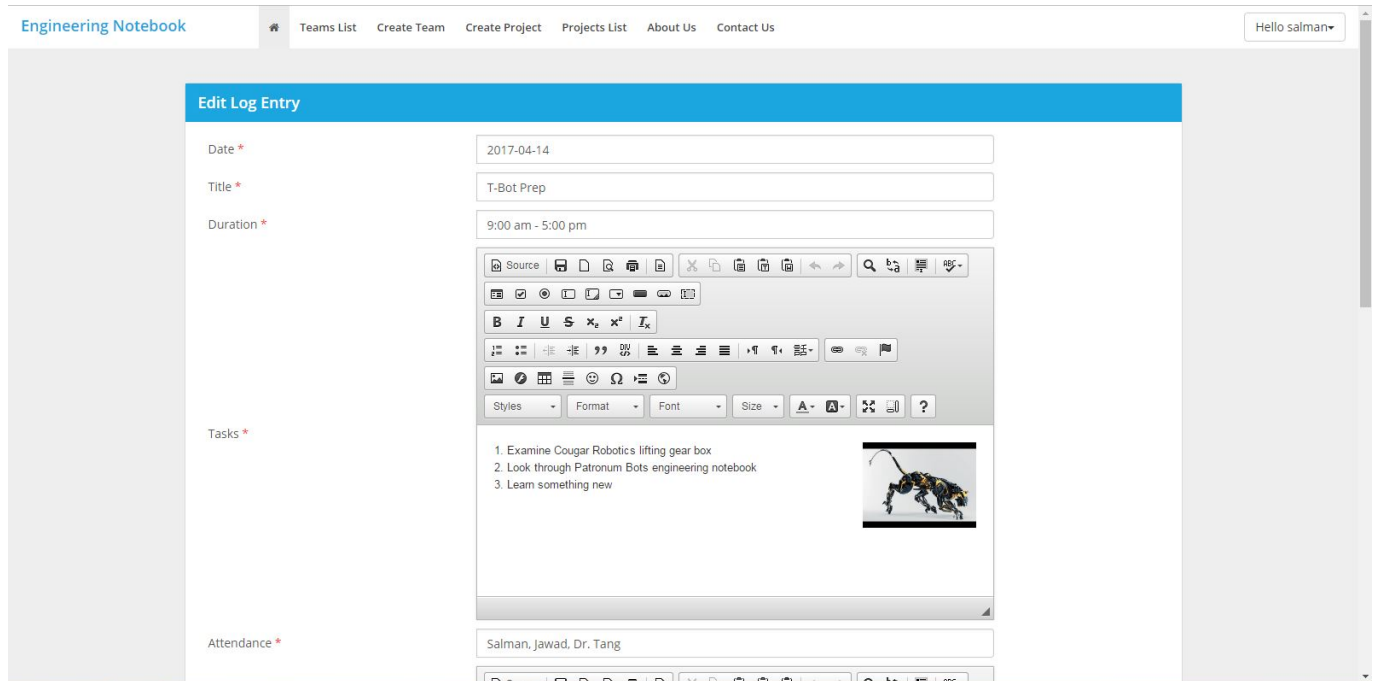


Figure 11 Captain Edit

Figure 12: Member Dashboard consists of projects list tab, teams list tab, about us tab and contact us tab. It also shows the name of the member at top right corner of the screen. The engineering notebook link on top left corner should take the user back to the homepage.

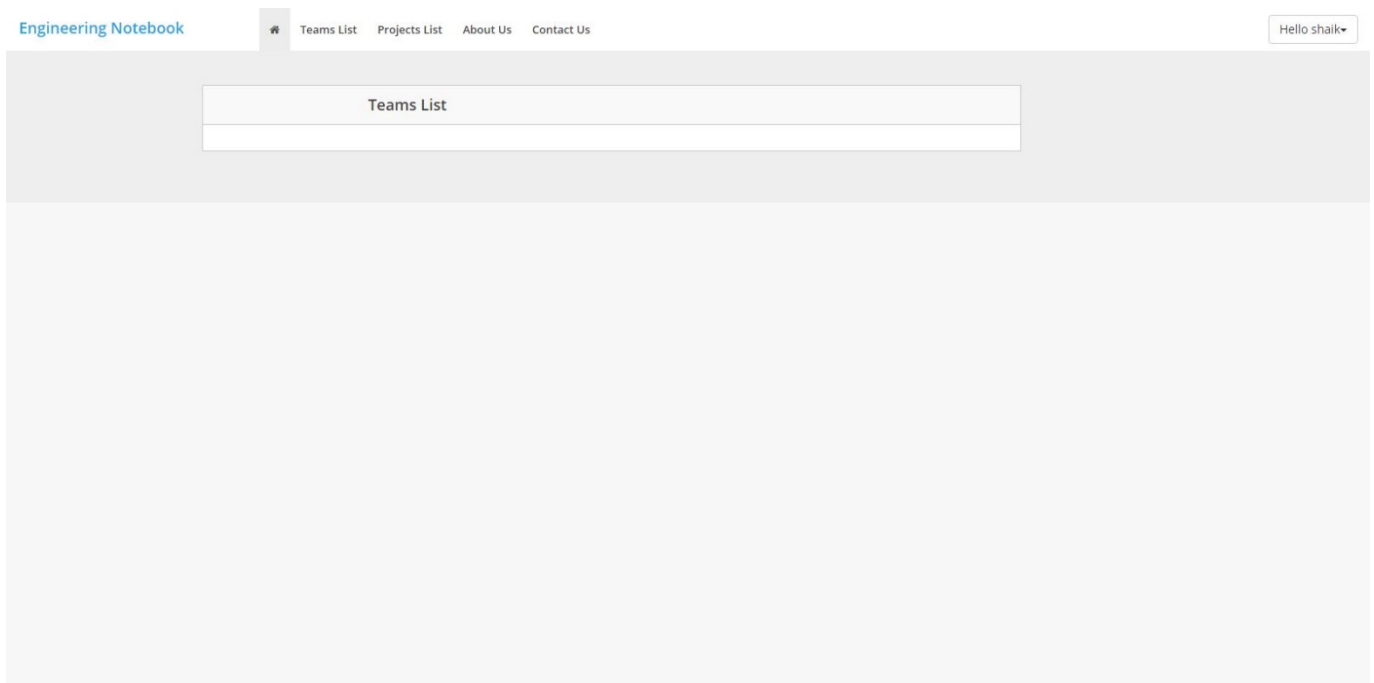


Figure 12 Member Dashboard

Figure 13: Member; Projects List. Shows all the existing projects in the system. Also, the search button allows the members to look for their desired projects. When they find the project they are looking for, they have the option to join by clicking on the send request button.

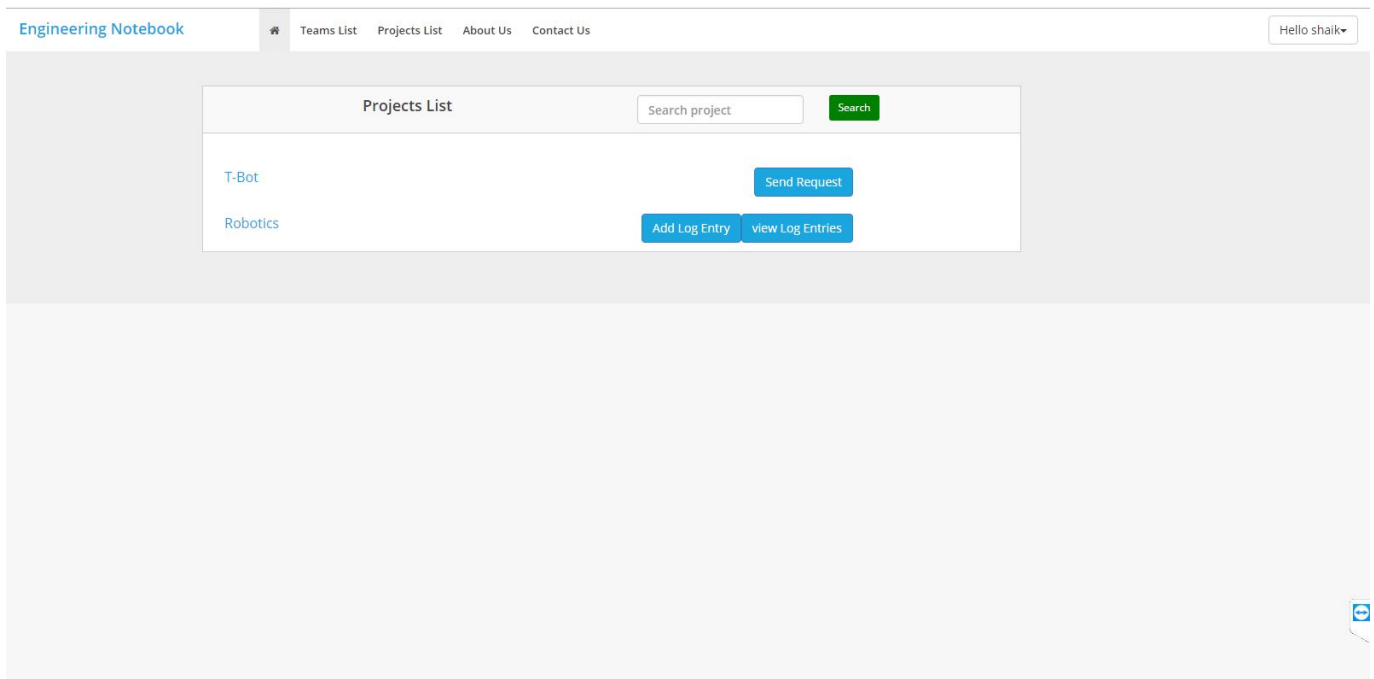


Figure 13 Member Projects List

Figure 14: Member; View Log Entries. This is one of the two features that a member gets to use in this web application. This allows the member to see the logs posted by his/her team members. Also, the search button at the top of the screen allows name wise search to look up logs entered by members.

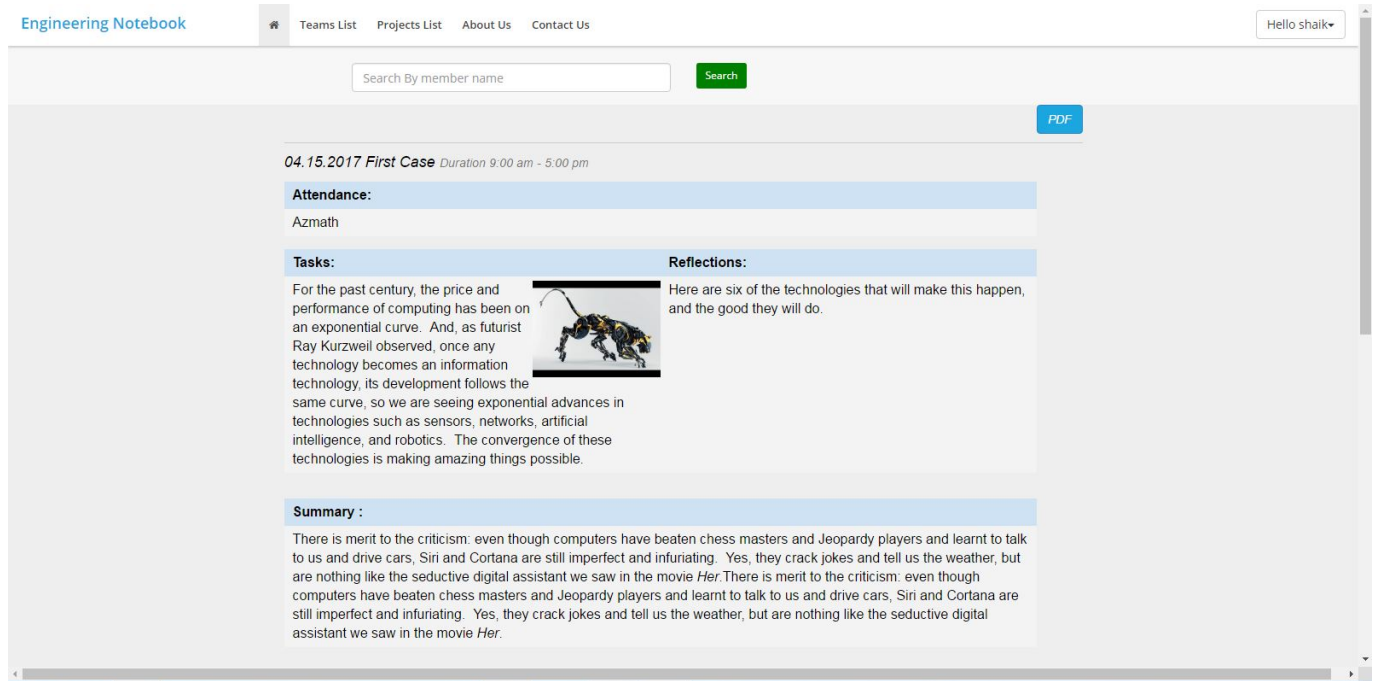


Figure 14 Member View Logs

Figure 15: Member; Add Log Entry page. This is the second feature that a member gets to use. Using this, a member can add logs to the project. The logs may be in the form of texts or media. After completing the log, the member should click on the add entry button at the bottom of the page to add that log to the project.

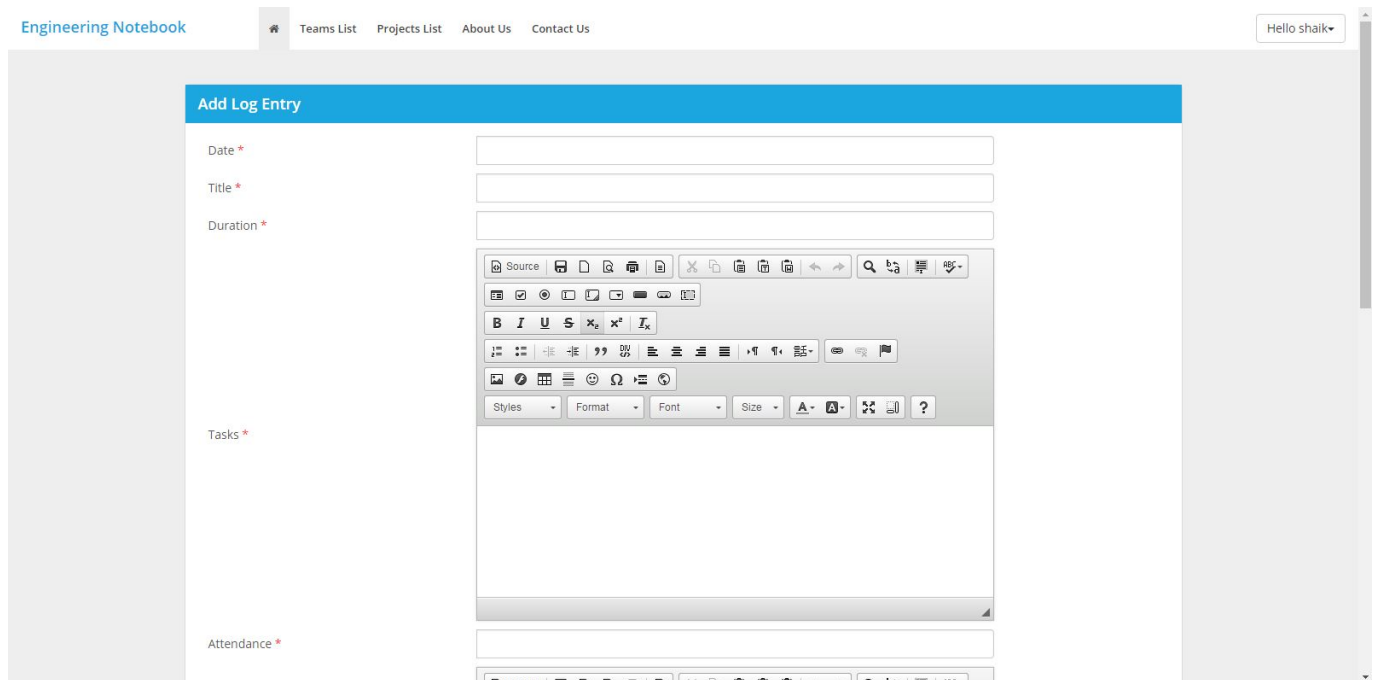


Figure 15 Member Add Logs

Figure 16: Search option in the logs allows the captain and the member to look up logs that other members were a part of. This is one of the most important requirements of this web application.

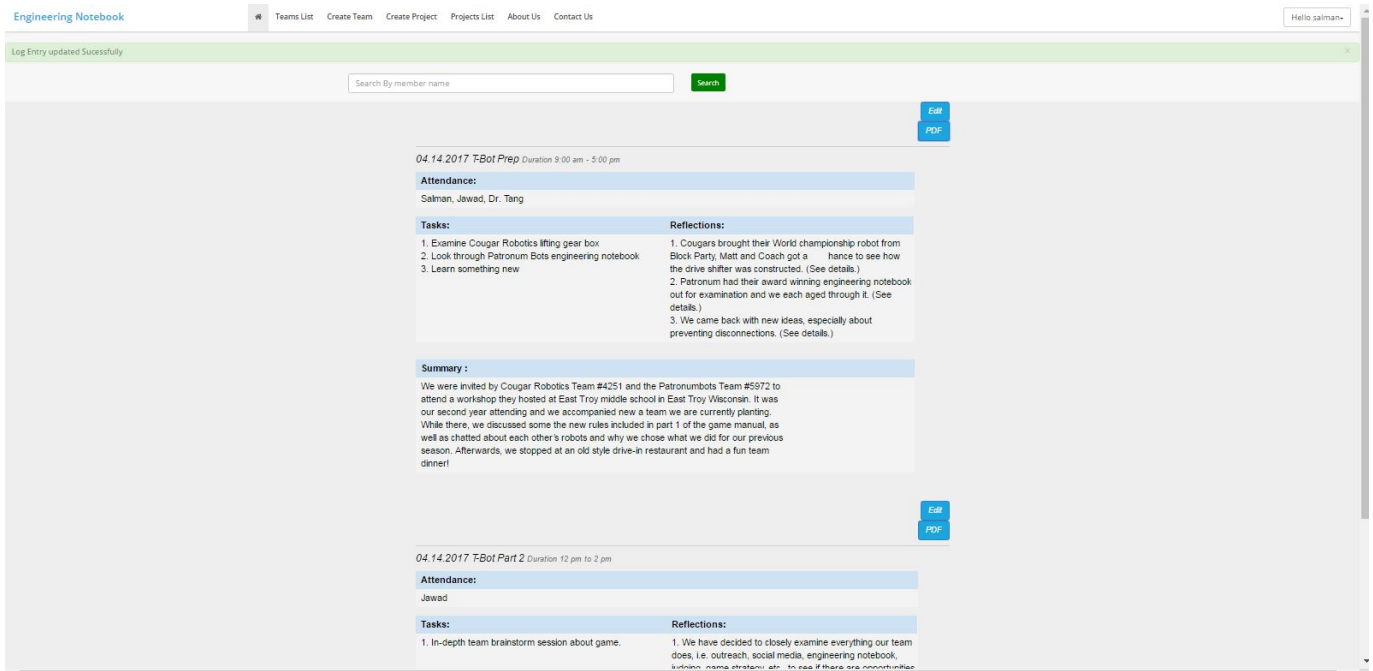


Figure 16 Search Feature

Figure 17: Conversion to PDF. Logs can be converted into PDF and downloaded by clicking on the PDF button. This is also one of the important requirement of this web application. The users such as the captain and the members should be able to use this feature.

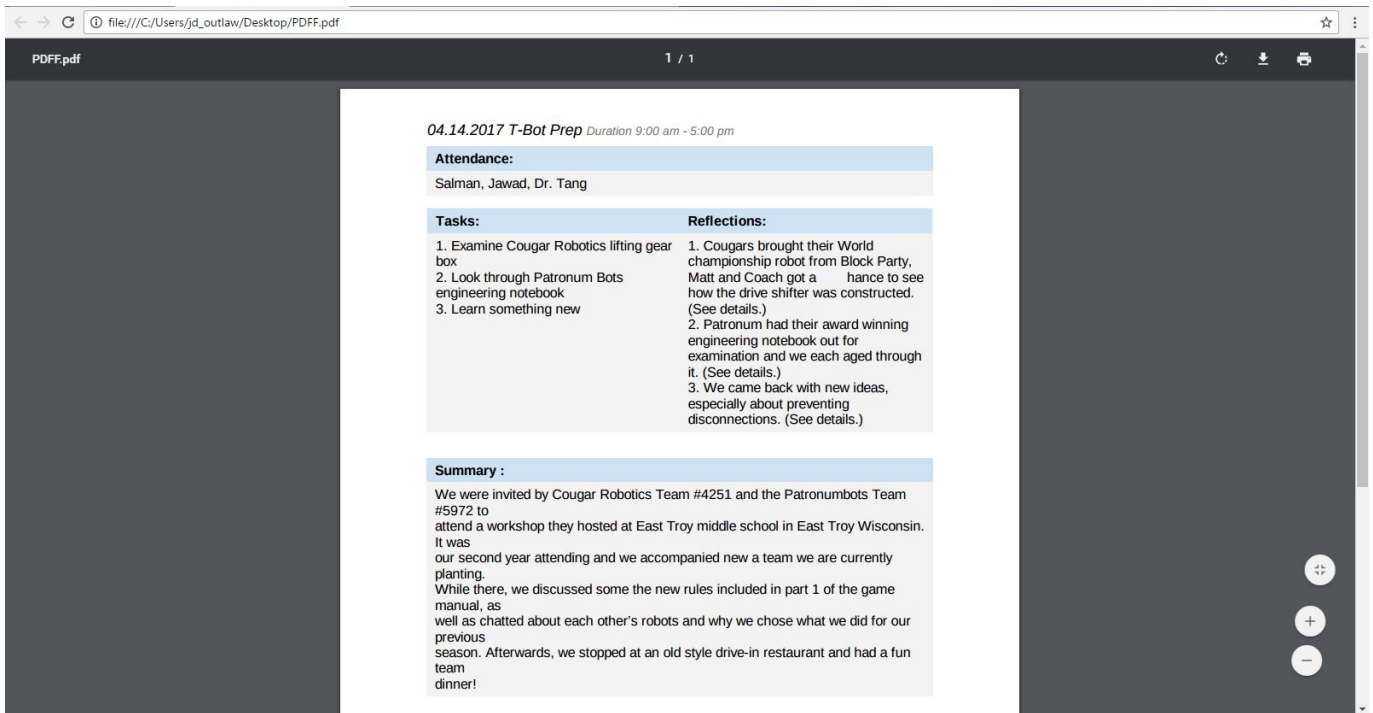


Figure 17 Convert to PDF

5 Internal/external Interface Impacts and Specification

Replace this section with a description of all internal and/or external interface changes. It should provide sufficient detail to begin updating user documentation, External Interface Specifications, and or Application Interface documents. Also describe here the impacts to any data structures that are shared across design units. If this Project design requires persistent data, describe the persistent data here at a high level.

6 Design Units Impacts

The 3 design techniques we used in this project are described below.

MDD:

Model Driven Development is used to make computer programs run faster. It is also known as model driven software development and model driven architecture. It lets individuals work as a team on projects even if their experience levels differ. MDD basically employs agile methodology to get the work done.

TDD:

Test Driven Design is the kind of software development where every part of source code is subjected to unit testing, making the code more reliable. Meaning, the unit testing is designed before the actual code is written. It focuses more on specification of user requirements than validation. Test Driven Design always helps build code that is usually bug free. It is agile in nature.

OOD:

Object Oriented Development encourages code reuse and provides organizations with competitive advantage, making the job easier for the users. It helps reduce maintenance. Object oriented systems are more flexible and reliable than other systems.

6.1 Functional Area A/Design Unit A

6.1.1 Functional Overview

The objective of the project is to build engineering notebook web application for the members of engineering project. By using this application, users should be able add their project information like text and graphical data to the engineering notebook. The team will have many team members and one captain with each team. The captain has an authority to activate and deactivate any member of their team. Also, admin has the power to remove any project for any reason. However, the members can't do these things, and they are able to add the information to their notebooks.

6.1.2 Impacts

The Impacts of the system are as follows:

- I. Members or users should have access to the system over the network.
- II. The captain should have authority to remove and accept any team member.
- III. The users should have the valid Email ID for sign up and sign in.

6.1.3 Requirements

The requirements for the successful implementation of web application includes as follows:

The web application should have three different kind of users with different authorities on the engineering notebook such as admin, captain, and member.

The users should be able to add, delete and update the project text and graphical information. Also, the application should allow the user to join as a captain or as a member. The captain should be able to activate or deactivate members of the project. The admin should have the privilege to accept a captain's request to create a project and delete it. The web application

should store huge amount of data, and it should handle heavy traffic. The users should be able to login, edit and update the information from any device like tablet, phone and desktop.

7 Open Issues

None.

8 Acknowledgements

We would like to thank Dr. Tang for letting us work on this project under her guidance. We acknowledge the fact that she motivated us to do better at every meeting we had. We thank her for being very convenient about the weekly meetings.

Last but not the least, we would like to thank Governors State University for giving us this opportunity and for being the school we needed it to be.

9 References

- [1] What is XAMPP? (n.d.). Retrieved April 26, 2017, from <https://www.apachefriends.org/index.html>
- [2] PHP 5 Tutorial. (n.d.). Retrieved April 26, 2017, from <https://www.w3schools.com/php/>
- [3] Banas, D. (Director). (2015, December 27). Install XAMPP [Video file]. Retrieved April 26, 2017, from <https://www.youtube.com/watch?v=xdvVKywGlc0>
- [4] Beginner PHP Tutorial - 1 - Introduction to PHP [Video file]. (2011, April 19). Retrieved April 26, 2017, from <https://www.youtube.com/watch?v=iCUV3iv9xOs&list=PL442FA2C127377F07>
- [5] Team Management Resources. (2017, April 4). Retrieved April 26, 2017, from https://www.firstinspires.org/sites/default/files/uploads/resource_library/ftc/team-5037.zip

10 Appendices

1. Admin- Administration/Administrator.
2. XAMPP- Cross-Platform (X), Apache (A), MySQL (M), PHP (P) and Perl (P).
3. FTP- File Transfer Protocol.
4. HTTP- Hypertext Transfer Protocol.
5. DB- Database.
6. MySQL- My Structured Query Language.
7. RDBMS- relational database management system.