DIYup

Team 206
Phyo Htut (phutut@mail.sfsu.edu) [Team Lead]
Eduardo Ramos (eramos4@mail.sfsu.edu) [QA/UI/UX]
Myles Pedronan (mpedrona@mail.sfsu.edu) [Backend Lead]
Antonio Carmona (acarmona@mail.sfsu.edu) [DB/Git Master]
Jason Wei (zwei@mail.sfsu.edu) [Frontend Lead]

Software Engineering
CSC 648/848 Section 4
Fall 2019

Version Table

Date	Milestone	Version
10/10/19	Milestone 1	1.0.1
10/3/19	Milestone 1	1.0.0

Table of Contents

Executive Summary

For most hobbyists, web-surfing for Do-It-Yourself (DIY) projects can sometimes be difficult since they are all scattered all over the internet. We believed that it would be nice if there were a large hub of DIY projects, and as a result, the idea for DIYup was born. With DIYup, guests and creators alike can browse and upload their own step-by-step DIY tutorials. Creator or not, our site makes it easy for anyone to pick up a variety of projects and follow step-by-step instructions with pictures to easily complete their project. Users can easily browse through the hottest projects on our site and sort depending on the difficulty or on the category of the project to suit their interests and level of building knowledge. Becoming a registered user comes with its own perks. As a registered user, users can become authors and upload their own projects, share their experience making a project from our site, rate, and save their favorite projects. As an author, authors can view and respond to comments to posts they've made as well as edit their original post in case of any typos or other errors. Any posts that are reported and found to be violating our site's rules will be deleted by site administrators. With site administrators there will be a team looking at reported posts to ensure users follow our site's rules and guidelines in order for our users to have the most enjoyable experience.

Use Cases

Defining Actors

Consider the task of finding actors for a website which provides DIY tutorials. The sites connects the authors and viewers.

Potential actors are:

Guest - A guest represents any user who is not logged into the site.

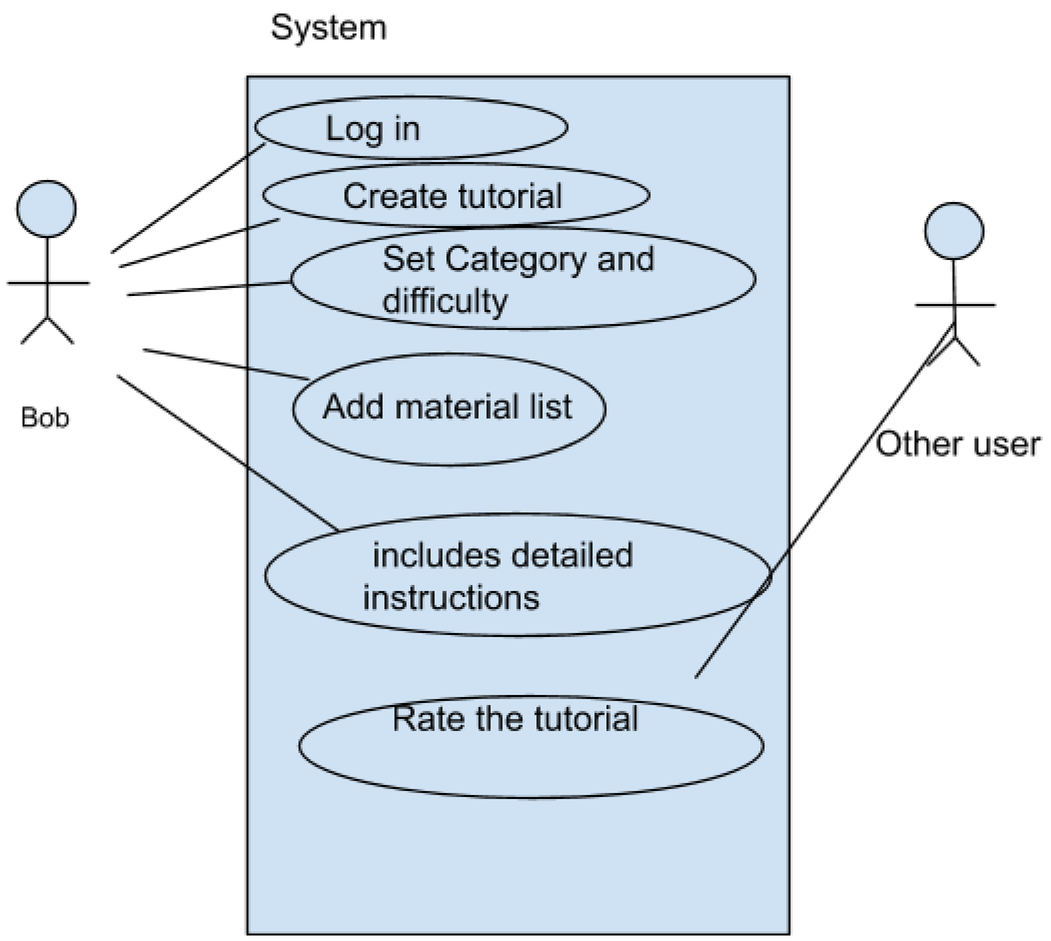
Registered user - A registered user represents any user who is logged into the site.

Author - An author represents a registered user who posted a tutorial on the site.

Admin - An admin represents person who manages the site.

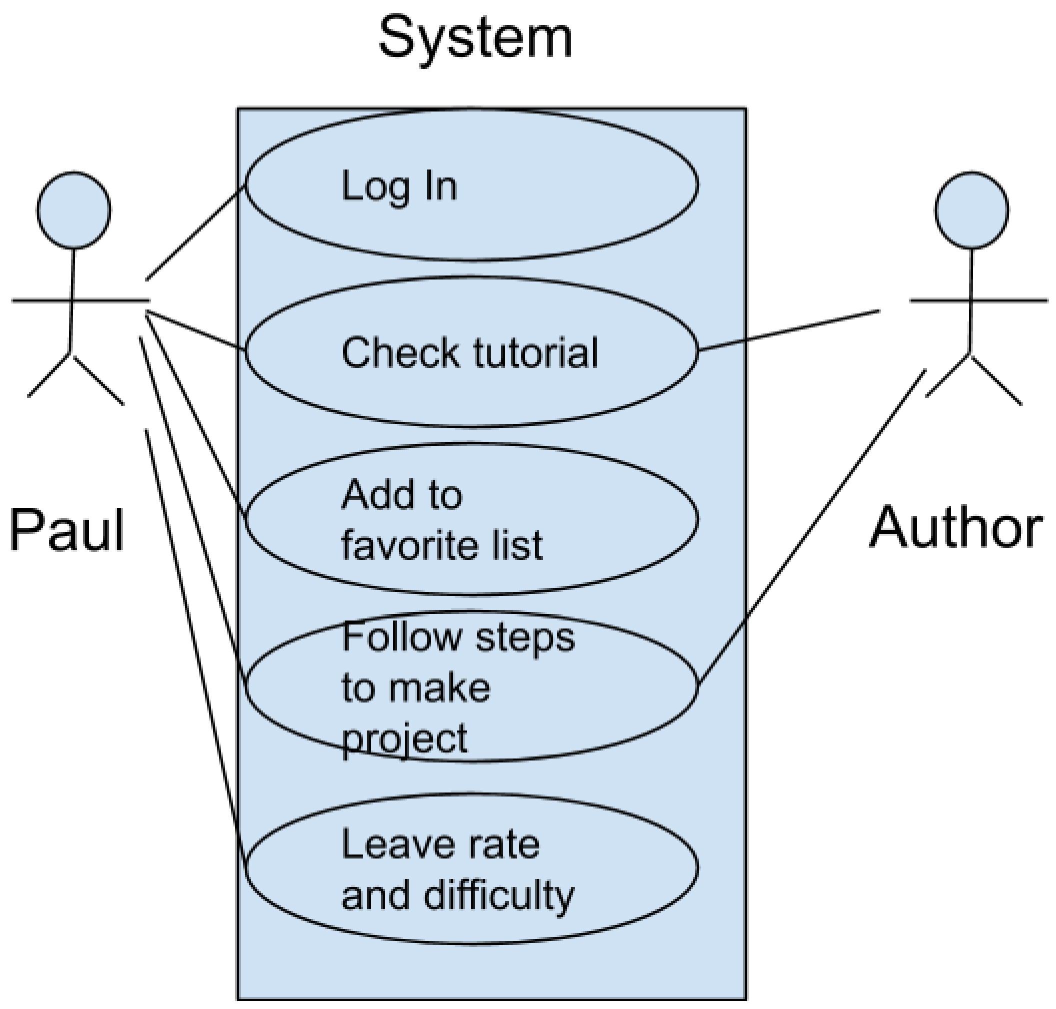
Actor: Bob - Author

Bob recently has a new DIY idea for making a rotating closet. Bob logs in the site and creates a new tutorial for his idea. Bob sets the tutorial's category to "Woodworking" and gives the project a difficulty. Bob adds a material list and tools list to his tutorial. After Bob finishes writing the tutorial, Bob includes detailed instructions with pictures for each step and submits the tutorial to the website. After a few days, Bob's rotating closet tutorial gets good ratings by other users and is shown on the "hot" page.



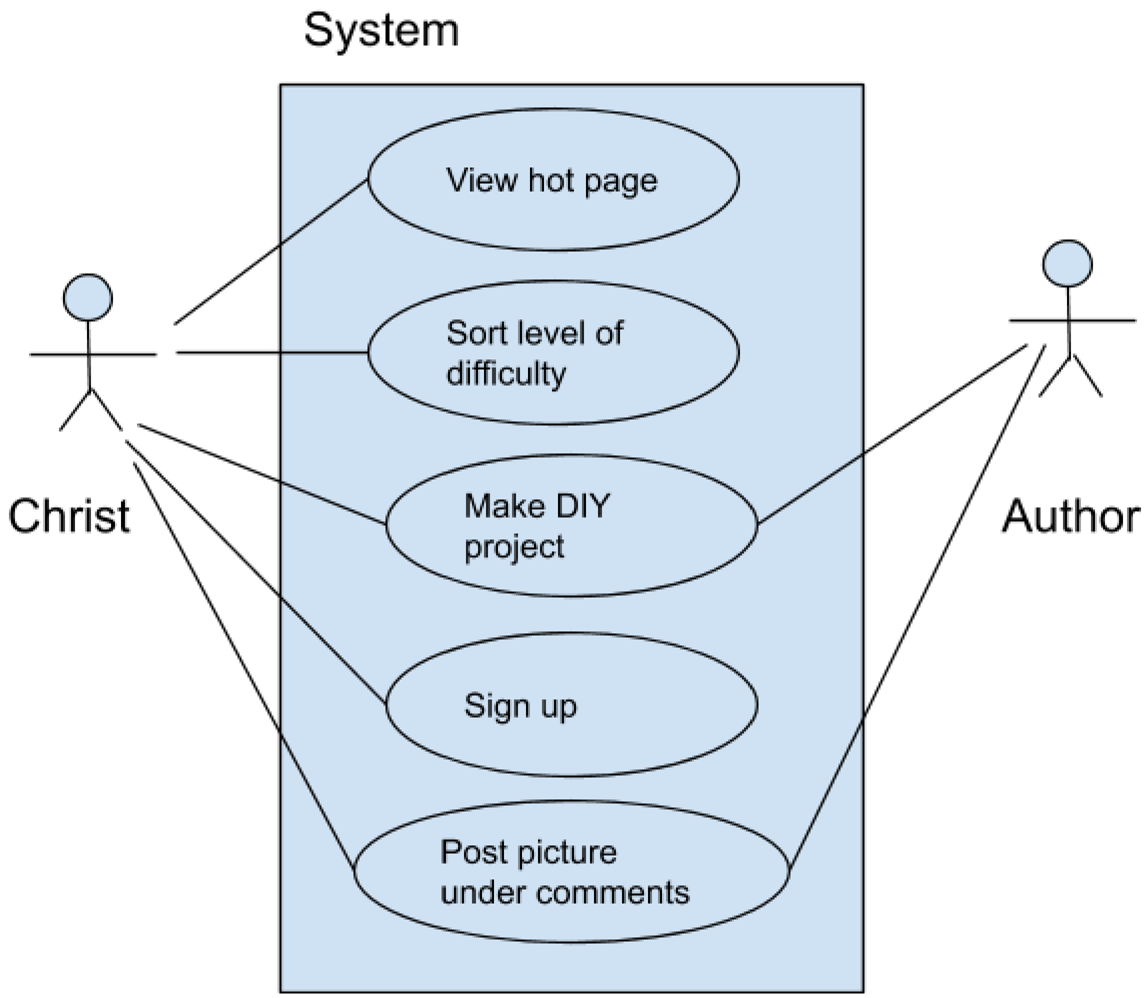
Actor: Paul - Registered User

Before a Paul decides what he wants to try for his first DIY project, he logs into our site and checks the tutorials that are shown in the “hot” page. Paul looks into a project on how to make a DIY Food Storage Shelf. Paul checks the material and tools lists and reads through all the steps. After looking over the category and difficulty of the tutorial, Paul feels confident enough to make it himself, so he adds it to his favorite list. Paul gets all of the necessary materials and tools from his local hardware store, follows all the steps that author posted, and completes the project. He returns to the tutorial, leaves a good rating and al difficulty score for the tutorial.



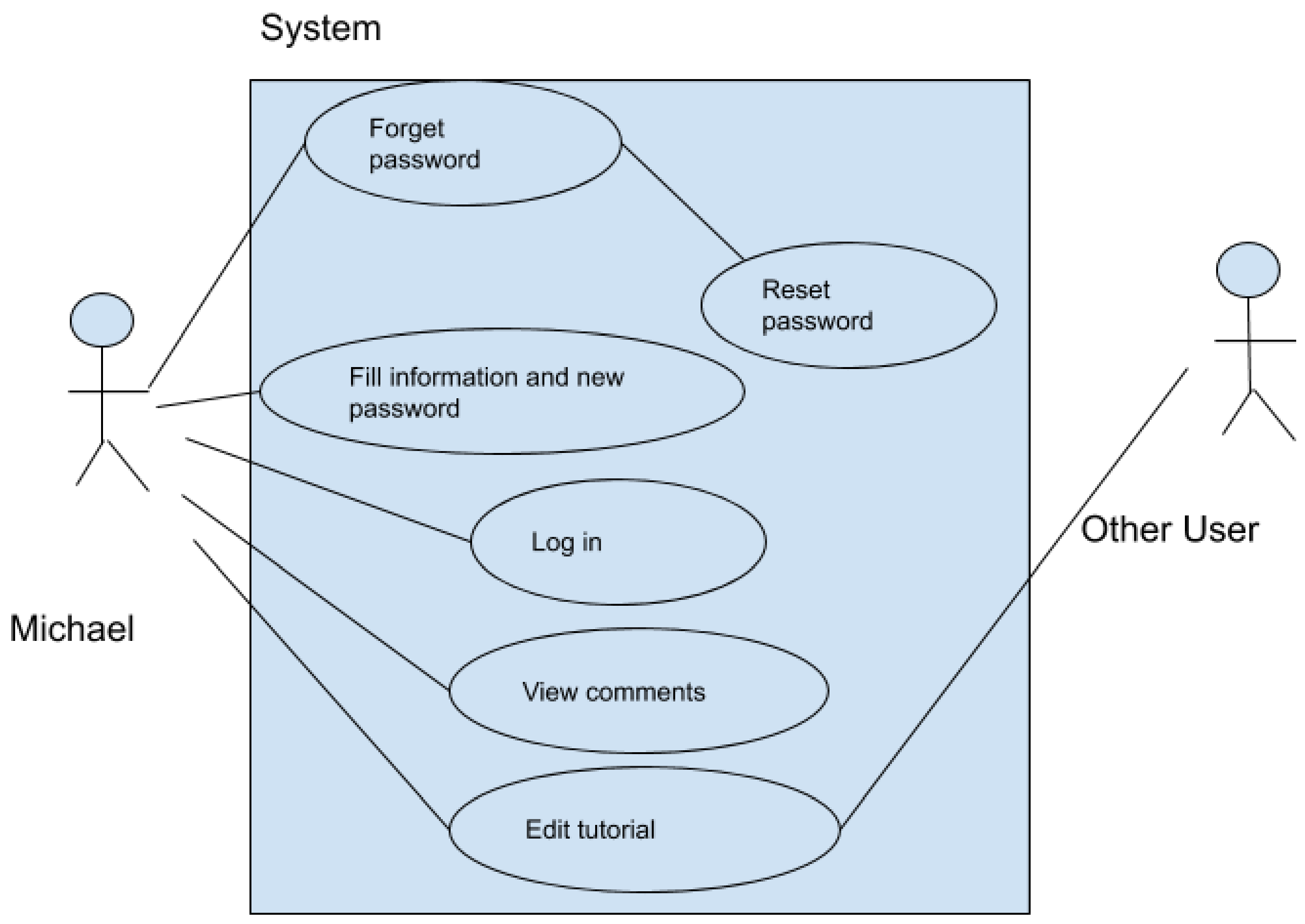
Actor: Christ - Guest

Christ recently heard about the site from his friend and he wants to try to make his own DIY project. Christ looks into the hot page, but he finds that most tutorials are too hard for a beginner, so he sorts them by level of difficulty. The guest believes that the “Fireflies in a Jar” tutorial was cool and easy for a beginner. Christ made his own DIY Jar within a few hours by following the steps. In order to share his picture, he had to sign up a new account. So Christ signs up for an account and posts a picture of his project in the comments section of the original tutorial.



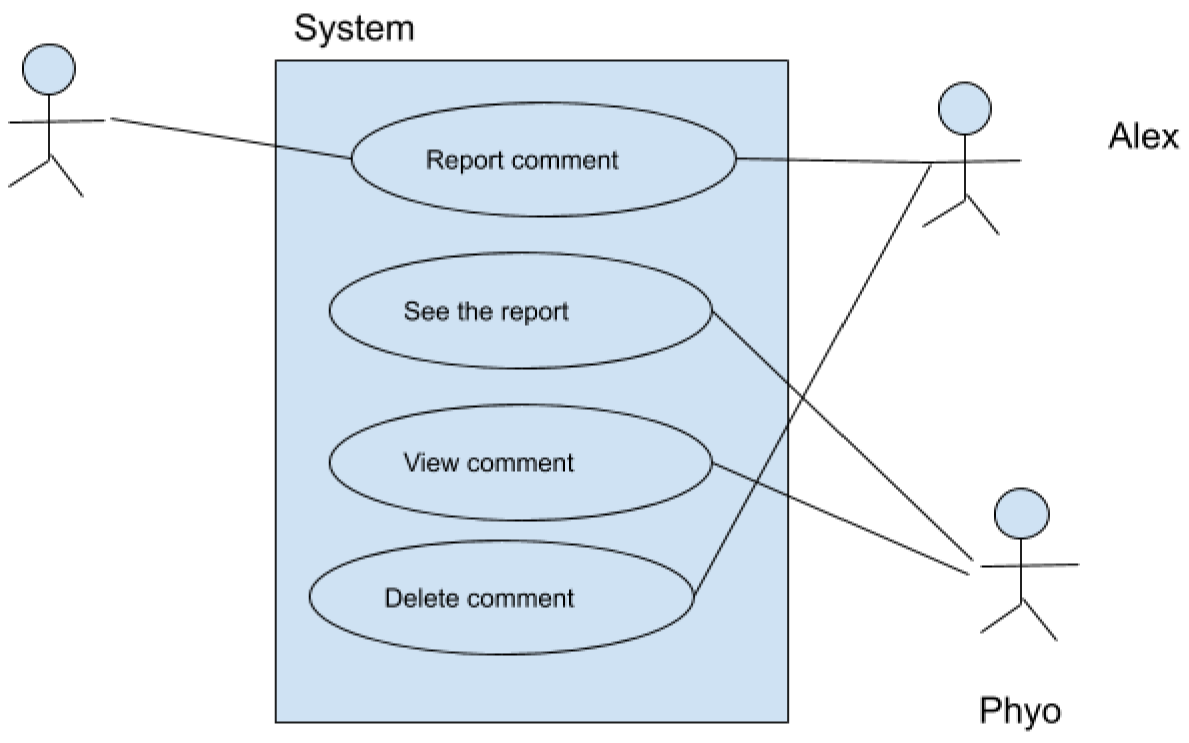
Actor: Michael - Author

Michael forgets his account's password after he gets a new computer. To retrieve the password, he clicks 'Forgot Password'. After Michael fills his information and new password, his password is reset. Michael logs in and goes to his post to view other user's comments and from user's comments, he realizes that he has a typo in his post. So Michael edits his post and comments to apologize for the confusion.



__Actors: Phyo - Admin, John, Alex - Registered User__

John saw comments under a bathroom DIY tutorial advertising another DIY website. So John reports Alex's comment. When the Phyo is checking if the site has any security issues, the admin sees the report from John. After the Phyo viewed the report, he believed that the comment is against the rules and deletes Alex's comment.



Data Definitions

- *Users*
  - **Registered User:** A user of the site that is logged in (and has an account). A ***Registered User*** has a unique *email address,* a unique *username*, an *avatar* image, and *password*.
  - **Guest User:** A user of the site that is not logged in (and may have an account). The information of a ***Guest User*** is not stored in any form.
  - **Admin:** A site administrator that is logged in (and has an account). An ***Admin*** has a unique *email address*, a unique *username*, and *password*.

- *Entities*
  - **Tutorial:** A tutorial posted by a ***Registered User***. A ***Tutorial*** is associated with the *email address* of a single ***Registered User*** and has a unique *ID*, a *title*, an *image*, a *category*, a *description*, an *author difficulty*, a *viewer difficulty*, a list of *materials*, a list of *tools*, and a *rating*.
  - **Step:** A single step of a ***Tutorial***. A ***Step*** is associated with the *ID* of a single ***Tutorial***, has a unique *ID*, some *content*, an *image*, and is associated with the *IDs* of the previous and next ***Steps*** in the ***Tutorial*** that it is associated with.
  - **List:** A list of items associated with a ***Tutorial***. A ***List*** is associated with the ID of a single ***Tutorial***, has a unique ID, a list of *contents*, and a list of *links*.
  - **Comment:** A comment posted on a ***Tutorial*** by a ***Registered User***. A ***Comment*** is associated with the *ID* of a single ***Tutorial*** and the *email address* of a single ***Registered User***. It has a unique *ID*, some content, an *image*, a *timestamp*, and is associated with the *IDs* of the previous and next ***Comments*** on the ***Tutorial*** that it is associated with.

- *Attributes*
    - **Category:** Used to describe the main area of work that a ***Tutorial*** is associated with. Several examples include Woodworking and Electronics.
    - **Difficulty:** Used to describe the amount of experience required in the specified **Category** associated with a ***Tutorial*** that is needed to complete the ***Tutorial***.
        - Author Difficulty: The difficulty of a ***Tutorial*** set by its author (a ***Registered User***).
        - Viewer Difficulty: The difficulty of a ***Tutorial*** set by ***Registered Users***.
    - **Rating:** Used to represent the quality of a ***Tutorial*** based on feedback from ***Registered Users***.

Functional Requirements

Guest User

1. Guest users shall be allowed to register for an account.
2. Guest users shall be allowed to log in if they have an account.
3. Guest users shall be allowed to search for tutorials posted by registered users.
4. Guest users shall be allowed to view tutorials posted by registered users.
5. Guest users shall be allowed to view comments posted by registered users.

Registered User

1. Registered users shall be allowed to log out.
2. Registered users shall be allowed to recover their account passwords.
3. Registered users shall be allowed to search for tutorials posted by registered users.
4. Registered users shall be allowed to view tutorials posted by registered users.
5. Registered users shall be allowed to view comments posted by registered users.
6. Registered users shall be allowed to post tutorials.
7. Registered users shall be allowed to include tools lists with their posted tutorials.
8. Registered users shall be allowed to include materials lists with their posted tutorials.
9. Registered users shall be allowed to edit their posted tutorials.
10. Registered users shall be allowed to delete their posted tutorials.
11. Registered users shall be allowed to post comments.
12. Registered users shall be allowed to edit their posted comments.
13. Registered users shall be allowed to delete their posted comments.
14. Registered users shall be allowed to rate tutorials posted by other registered users.

15. Registered users shall be allowed to provide difficulties to tutorials posted by other registered users.
16. Registered users shall be allowed to report tutorials posted by other registered users.
17. Registered users shall be allowed to report comments posted by other registered users.
18. Registered users shall be allowed to add tutorials posted by other registered users to a personal favorites list.
19. Registered users shall be allowed to remove tutorials posted by other registered users from a personal favorites list.

Admin

1. Admin accounts shall be reserved to the developers.
2. Admin accounts shall be created on the server side.
3. Admins shall have all of the abilities of registered users.
4. Admins shall be allowed to delete any tutorials.
5. Admins shall be allowed to delete any comments.

Non-Functional Requirements

Performance & Reliability Requirements:

- The site's UI shall respond visually within 5 seconds across all pages.
- All database tables shall use indexes to speed up queries.
- The site shall have an hourly uptime percentage of at least 80%.

Storage & Media Requirements:

- User images shall be uploaded to Imgur.
- Images uploaded by users shall be stored in the database as Imgur links.
- Users shall be allowed to upload JPEG and PNG format image files.
- The maximum uploaded image file size shall be 2MB.

Privacy & Security Requirements:

- Data that the site shall collect includes the email addresses of registered users and pictures uploaded by registered users.
- The email addresses provided by registered users shall only be visible to their associated users and DIYup admins.
- Users shall be required to validate their email addresses before logging in for the first time.
- User passwords shall be stored in the database after being hashed.

Compatibility Requirements:

- The site shall be compatible with Chrome 51 and above.
- The site shall be compatible with Firefox 54 and above.
- The site shall be compatible with Microsoft Edge 14.0 and above.
- The site shall be compatible with Safari 10 and above.
- The site shall be compatible with Opera 38 and above.

Marketing Requirements:

- The site’s logo shall be present somewhere on every page of the site.
- The name of the site shall always be stylized as “DIYup” when referenced..

Coding Standards:

- No line of code shall have more than 80 characters.
- No functions shall have more than 80 lines of codes.
- Helper functions shall never be declared before the primary function.
- Variables should be self-explanatory.
- No variables shall be unused.
- No functions shall be unused.
- All functions must be thoroughly explained.
- All error messages shall be verbose.

Competitive Analysis

Feature / Company	Makezine	Instructables	DIY Network	wikiHow	DIYup
Community Rating	-	-	-	+	+
Save Favorites to User Profile	-	+	-	-	+
Comments from registered users	-	+	-	-	+
Difficulty Rating	+	-	+	-	++
Parts/Tools Lists	+	-	+	+	++
Project Prices	+	-	+	-	-

With DIYup we plan to provide the most useful way for users to find projects that they enjoy and are suited to their building expertise. Like our competitors, we provide a large selection of projects to suit users' interests. Unlike other competitors, we focus on the core features that will make the user experience as easy to use and seamless as possible. We plan on putting a higher emphasis on user experience by allowing more community involvement such as allowing users to rate, save and share their favorite projects. Users are able to add a community rating that can help better rate the difficulty of the project. We also have authors that can also include a list of parts to help aid users wanting to build the project and help them find all the materials they need. Guests and users visiting our site can find easily find something that they could enjoy building without the hassle of looking through a complicated website to find something they like. With having such an emphasis on our core features, users will have the best user experience despite the lack of features from other competitors.

High-Level System Architecture and Technologies Used

- **Sever:** *AWS micro 1 CPU 1GB RAM*
- **Operating System:** *Ubuntu 16.0.4*
- **Database:** *MySQL 5.7.27*
- **Web-Server:** *Apache 2.4.29*
- **Server-Side Language:** *Python 3.6.8*
- **Additional Technologies:**
    - **Frontend Framework:** *Quasar 1.1.1*
        - *Vue 3.7.0*
    - **Web Framework:** *Flask 1.1.1*
    - **IDE:** *VSCode 1.38.1*

Team

- *Phyo Htut (Team Lead)*
  - As a team lead, Phyo maintains the morale and welfare of all the team members as well as software deployment onto the cloud. He also serves as a satellite frontend and backend engineer to support Jason Wei and Myles Pedronan.
- *Antonio Carmona (Database and Git Master)*
  - As a database and git master, Antonio is in charge of version control as well as data flow for the software. He also serves as a satellite backend engineer to support Myles Pedronan.
- *Myles Pedronan (Backend Lead and Document Master)*
  - As a backend lead, Myles is in charge of creating RESTful APIs so that the client side can retrieve server side data from the server. He also serves as a document master, which means he edits and reviews all the documents of the group.
- *Eduardo Ramos (UI/UX and QA Tester)*
  - As UI/UX and QA tester for the software, Eduardo creates the layout of the program as well as making sure all the components are functionally correct when the events are triggered. If bugs were found, he will report the bugs to Phyo (Team Lead) and Jason (Frontend Lead) in order for the bugs to get resolved.
- *Jason Wei (Frontend Lead)*
  - As a frontend lead, all the client side behaviors and software layouts are implemented and managed by Jason. He works closely with Eduardo and Phyo in order to improve the looks and feels of the program.

Checklist

Item	Status
Team found a time slot to meet outside of class	**DONE**
Github master chosen	**DONE**
Document master chosen	**DONE**
Backend Lead chosen	**DONE**
Frontend Lead chosen	**DONE**
UI/UX and QA chosen	**DONE**
Team decided and agreed together on using the listed software tools and deployment server	**DONE**
Team ready and able to use the chosen backend and frontend framework	**DONE**
Team lead ensured that all team members read the final M1 and agree/understand it before submission	**DONE**
Github organized as discussed in class (eg. master branch, development branch, folder and milestone documents etc.)	**DONE**