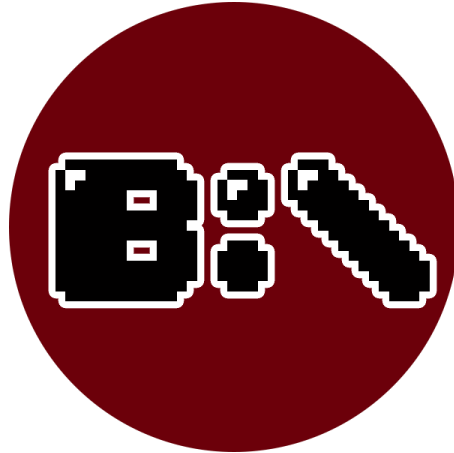


TE³D BUNGALOW

Software Requirements Specifications



Team Bictory

BORROMEO, Klaudia Gaia
CASTRO, Joesei Jesus
CRUZ, Edmund Gerald
GOMEZ, Arlan Ross
LAGAYAN, Jan Alysa
MARQUEZ, Samuel Cua
NOBLE, Neil Michael
TAN, Aron Joshua
TANCHUAN, Rafael Louis

Table of Content

I. Preface	1
1. Purpose	1
2. Scope	1
II. Introduction	1
1. Background	1
2. Overall Description	1
III. Glossary	2
IV. Product Perspective	3
V. Product Function	4
VI. User Characteristics	4
VII. Limitations	5
VIII. Assumptions and Dependencies	5
IX. Apportioning of Requirements	5
X. External Interface	6
XI. Specific Requirements	6
1. Functions	6
2. Usability Requirements	11
3. Performance Requirements	12
XII. Appendix	13
XIII. Index	16

Preface

1. Purpose

The goal of this project is to provide a web-based repository for manuscripts, journals, codes, and theses under TE³D House, which stands for Technology, Education, Entertainment, Empathy, and Design House, a research laboratory based in DLSU-M Laguna Campus.

2. Scope

The repository will only store projects of TE³D House, only TE³D House members and other collaborators can work on the project.

Introduction

1. Background

TE³D House is a research laboratory based in DLSU-M Laguna Campus. The laboratory has several ongoing, finished and planned projects. Because of this, there is a need for a specific web application, a repository that would let the TE³D House members keep track of and store their projects for convenience in the further progress of TE³D House.

2. Overall Description

The web application is a repository for projects. The repository lets users add, edit and delete projects, and view and retrieve the files of projects. The repository is needed because TE³D House needs a digital storage for their projects such as manuscripts, source codes, journals, and theses. This will be available to TE³D House members and their collaborators. Those not affiliated with TE³D House can also use the application, but can only view project title and abstract and may contact the project head for further inquiries regarding the project.

Glossary

Access	a set of features available based on the user type.
Add	the action of creating a project in the repository.
Admin	short for administrator; user who runs and maintains the website and repository.
Application	a program or piece of software designed and written to fulfill a particular purpose of the user.
Bungalow	the name of the repository; based on a type of house that has one storey building with a low-pitched roof.
Collaborator	companies and organizations in collaboration with TE ³ D House.
Database	a collection of information that is organized so that it can be easily accessed, managed and updated.
Delete	the action of removing a project and its information from the database on the web server.
DLSU-M	De La Salle University-Manila.
Internet	a global computer network providing a variety of information and communication facilities, consisting of interconnected networks using standardized communication protocols which will be a dependent factor of this application's functionality.
Edit	to change or correct any component of a project.
Fellow	faculty and/or mentors.
File	data uploaded into a project by admins or members. These could be the following, but are not limited to, .java, .c, .txt, .docx, .xlsx, etc.
Guest	users of Bungalow who do not own an account.
Journal	a newspaper or magazine that deals with a particular subject or professional activity.
Member	users of Bungalow who have an account, members of TE ³ D House and/or are working on a TE ³ D House project.

Permission	authorization to use a feature of the application.
Project	1. it is what is added into the repository, and contains three sections: abstract, files, and members which can all be edited. 2. To be used as a general term for thesis, journal, and the like.
Repository	a central place in which an aggregation of data is kept and maintained in an organized way, usually in computer storage.
Retrieve	acquire a downloaded copy of the files from latest update of the project from the repository.
TE³D House	stands for Technology, Education, Entertainment, Empathy, and Design House, a research laboratory based in DLSU-M Laguna Campus.
Thesis	a long essay or dissertation involving personal research, written by a candidate for a college degree.
Upload	to transfer data into the database.
Users	a person who uses or operates the web application.
View	to be able to see more than the project title and abstract which includes the files and members section.

Product Perspective

Bungalow is a web application that requires an Internet connection and an account to access resources from the database which stores the projects that TE³D members may add. Bungalow has three types of users that have different permissions with database access. Users can search for projects, in addition to that, they may use various filters and categories for more specific results.

The web application and the database server constantly communicate with each other to provide each other with the necessary resources. The web application is reliant to network connectivity and cannot maintain its normal function without it.

Product Functions

The following are the primary functions of the application:

- 1. Repository**

Users with the appropriate permissions will be able to add projects, delete and edit their own projects, and view and retrieve their own or other's' projects.

- 2. Permissions**

All the project files in the repository are private, available only to the members of the project team. Members of the Bungalow community whom are not part of the project team could ask permission for access to the files through the project head. Users that are guests cannot view and retrieve the files through Bungalow.

- 3. Advanced Search**

Users can search for projects that are already stored in the repository with at least one of the parameters namely title, members, date started, funded by and status (Finished, Ongoing or Unfinished).

- 4. Notifications**

Project heads are notified when they are being contacted by guests.

- 5. APA Documentation Generator**

Users can generate an APA Documentation of the product for citing.

User Characteristics

The following are types of users whom would be actively using the application on its release:

- 1. Admin**

Manages the projects and community of Bungalow. Can add, delete and edit projects, view and retrieve project files, receive notifications from projects they head in when guests are trying to contact them, create and terminate members.

- 2. Member**

A member of the community of Bungalow. Can add, delete and edit their own projects, view and retrieve project files, and receive notifications from projects they head in when guests are trying to contact them.

- 3. Guest**

Can contact the project head for project files. Cannot create their own projects. They can only view the abstract and title of a project.

Limitations

On release, the application will not be able to perform the following functions compared to other existing similar applications:

- 1. Conversations Thread**

A chatroom for messaging among Bungalow members will not be available.

- 2. Modification History**

A view of previous modifications or edits to projects will not be available.

- 3. Modification Notification**

Members of a project will not be notified if there were modifications to the project.

- 4. File Viewing**

The users may only view the files if they download them, which they may do if the account they are using has the appropriate permissions to retrieve them. The users cannot view files within the application prior to downloading a file.

- 5. File Content Editing**

The users cannot edit the files inside the application since it does not support any editing platform whatsoever, so the user can only reupload the file to keep it up to date.

Assumptions and Dependencies

On the release of the application, the following are assumed to be present and/or available to support the application's functions:

- 1. Network Connectivity**

The Internet connection is stable, can load the pages and execute functions (ie. downloading, searching, uploading, etc.) without taking too much time.

- 2. Browser Support**

The browser is compatible and can handle HTML elements of the application.

Apportioning of Requirements

In the future, the following could potentially be implemented to later versions of the application:

- 1. Conversations Thread**

A chatroom for messaging among Bungalow members.

- 2. Modification History**

There would be a view for previous modifications or edits on the contents of a project.

- 3. Modification Notification**

The members will be notified if there were new modifications to the project they are in.

4. File Viewing

A preview of files and other contents would be available in the application.

5. File Editing

Project owners would be able to edit the files and contents without downloading the updates.

External Interfaces

The application will take input from the user through elements in the web application, and will be connected to the database through PHP. The application will provide an interface for members to store their projects on the repository. Guests cannot retrieve the project files through Bungalow but can contact the project head for the files.

To use the application, users may choose whether or not to log in. Logging in and the type of account the user is logging in with affects the activities the user is able to do during their session (refer to Functional Requirements 1.3 and 2.2). Projects may then be added, edited, deleted, viewed, or retrieved depending on the access given to the user.

Specific Requirements

Functions

-User Class 1- Admin

Functional Requirement 1.1

ID: FR1

Feature: Manage Projects

In order to keep track of all projects, an Admin is able to manage his or her projects.

Scenario: Add a Project

When the admin creates a new project
Then the new project should be added

Scenario: Edit the content of an existing Project

When the admin edits an existing project
Then the project information should be updated

Scenario: Delete an existing Project

When the admin deletes an project
Then the project is deleted
And the project is no longer viewable or retrievable

Scenario: Finish a Project

When the admin selects done
Then the project status should change to finished

Scenario: Discontinue a Project

When the admin selects discontinue
Then the project status should change to unfinished

Scenario: Continue a Project

Given the project status is unfinished
When the admin selects continue
Then the project status should change to ongoing

Scenario: Add Members to Project

They can add members
To be part of the project

Functional Requirement 1.2

ID:FR2

Feature: Manage Members

In order to keep track of members
an admin should be able to manage all of the members.

Scenario: Create a Member account

When a person wants to become a member
Then the person shall give his/her details to the Admin
And the admin creates or recreates the account for said person

Scenario: Terminate a Member

When the admin terminates a member
Then the member is deleted
And his/her unfinished works are marked unfinished

Functional Requirement 1.3

ID: FR3

Feature: Admin log-in

In order to administer the system
An admin
Should be logged in

Scenario: Successful log-in

Given the admin wants to log in

When the admin logs in with an admin account

Then the admin should be logged in as an admin

Functional Requirement 1.4

ID: FR4

Feature: Search

In order to see the desired project

An admin

should be able to search

Scenario: Keyword Search

Search results should be viewed in a list depending on the searched keyword. Each element in the list represents a specific project. Each element should include the project name, project head, a short description, status of project, start date of project, funded by or last update of project.

Scenario: Tag Search

Search results should be viewed in a list depending on the tag selected. Each element in the list represents a specific project. Each element should include the project name, project head, a short description, status of project, start date of project, funded by or last update of project.

-User Class 2- Member**Functional Requirement 2.1**

ID: FR5

Feature: Manage Projects

In order to keep track of their projects

a member is able to manage his or her projects.

Scenario: Add a Project

When a member creates a new project

Then the new project should be added

Scenario: Edit the content of an existing Project

When a member edits an existing project

Then the project information should be updated

Scenario: Delete an existing Project

When a member deletes his/her project
Then the project should be deleted

Scenario: Finish a Project

Given the member is the project head of the project
When the member selects done
Then the project status should change to finished

Scenario: Discontinue a Project

Given the member is the project head of the project
When the member selects discontinue
Then the project status should change to unfinished

Scenario: Continue a Project

Given the member is the project head of the project
And the project is discontinued
When the member selects continue
Then the project status should change to ongoing

Scenario: Add Members to Project

Given the member is the project head of the project
They can add other members
To be part of the project

Functional Requirement 2.2

ID: FR6

Feature: Member log-in

In order to use the system
A member
Should be logged in

Scenario: Successful log-in

Given the member wants to log in
When the member logs in with his/her account
Then the member should be logged in as a member

Scenario: Retrieve Password

Given the member wants to log in

And has lost the password

When the member enters his/her email address in the “Retrieve password” form And submits the form

Then the member should receive an email containing the password

Functional Requirement 2.3

ID: FR7

Feature: Search

In order to see the desired project

A member

should be able to search

Scenario: Keyword Search

Search results should be viewed in a list depending on the searched keyword. Each element in the list represents a specific project. Each element should include the project name, project head, a short description, status of project, start date of project, end date or last update of project.

Scenario: Tag Search

Search results should be viewed in a list depending on the tag selected. Each element in the list represents a specific project. Each element should include the project name, project head, a short description, status of project, start date of project, end date or last update of project.

-User Class 3- Guest**Functional Requirement 3.1**

ID: FR8

Title: Ask for Project Content

DESC: A guest should be able to contact the project head to ask for project files or for more information about the project.

Functional Requirement 3.2

ID: FR9

Feature: Search

In order to see the desired project

A guest

should be able to search

Scenario: Keyword Search

Search results should be viewed in a list depending on the searched keyword. Each element in the list represents a specific project. Each element should include the project name, project head, a short description, status of project, start date of project, end date or last update of project.

Scenario: Tag Search

Search results should be viewed in a list depending on the tag selected. Each element in the list represents a specific project. Each element should include the project name, project head, a short description, status of project, start date of project, end date or last update of project.

Functional Requirement 3.3

ID: FR10

Feature: Contact

In order to allow guests to retrieve files
The project head of the project
Should be able to receive a message to
Share the file through external applications

Scenario: Contact for File Acquisition

Given that a guest contacts the project head
When the guest messages the project head through Bungalow
The guest may be able to receive the files through an external application
And the guest would be notified in their email

Usability Requirements

1. The repository should allow the users to understand it with minimal effort. Otherwise we cannot expect the users to switch to our system. In particular, success with downloading or viewing a project from our repository within the first few attempts is important. In this case, all types of users must be able to do their tasks without complications.

Usability Requirement 1.1

Users that are guests must be able to view the projects' title and abstract without complication. There should be an obvious and discrete interface for viewing and searching.

Usability Requirement 1.2

Users that are Members must be able to view, upload and download projects that they're allowed to. There should be an obvious and discrete interface for viewing and searching.

Usability Requirement 1.3

Users that are admins must be able to view, upload and download all projects. Admins are contacted by the guests through an off-application email.

2. The use of the search module must be simple and intuitive for the users. In this case, certain filters and advance search must be reasonably available for all kinds of users.

Usability Requirement 2.1

Users that are guests must be able to filter through projects by author, status, date, funded by and name. There should be an obvious and discrete interface for setting parameters for searching.

Usability Requirement 2.2

Users that are Members must be able to filter through projects by author, status, date, funded by and name.

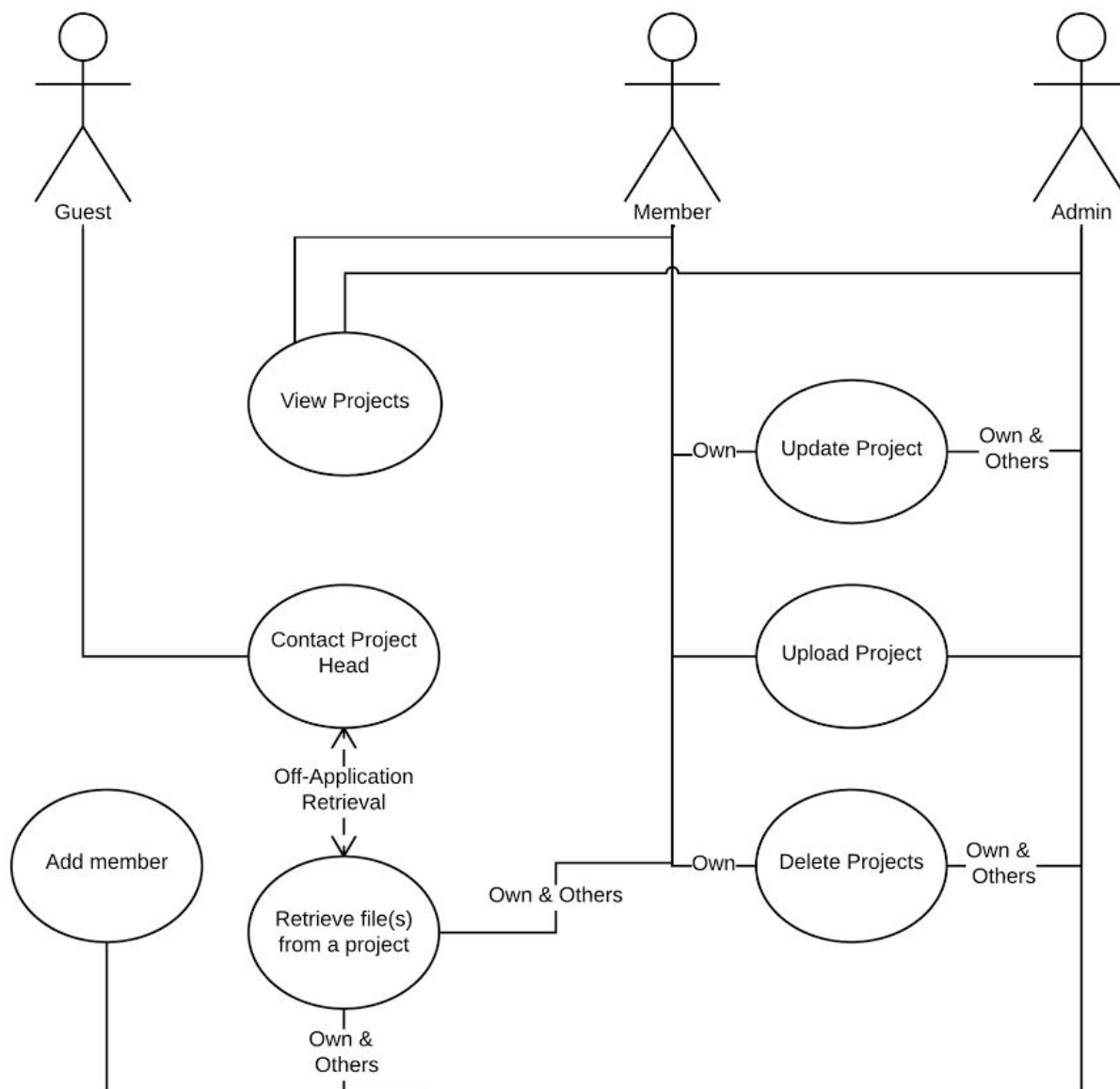
Usability Requirement 2.3

Users that are Admins must be able to filter through projects by author, status, date, funded by and name. There should be an obvious and discrete interface for setting parameters for searching.

Performance Requirements

Performance should not be an issue because our software will only be handling small queries. Changing screens should happen quickly if not instantaneously. Uploading and downloading on the other hand may have varying performances but is only defined by the strength of the device's connection to the Internet.

Appendix Use-Case Diagram



User-Function Relationship

Legend:

A - Available

U - Unavailable

NA - Not applicable

Uploading Projects	Admin	Member	Guest
Own Projects	A	A	U

Updating Projects	Admin	Member	Guest
Own Projects	A	A	NA
Others' Projects	A	U	U

Deleting Projects	Admin	Member	Guest
Own Projects	A	A	NA
Others' Projects	A	U	U

Viewing Projects	Admin	Member	Guest
Own Projects	A	A	NA
Others' Projects	A	A	NA

Retrieve Content	Admin	Member	Guest
Own Projects	A	A	NA
Others' Projects	A	A	NA
Ask for Content	NA	NA	A

Membership	Admin	Member	Guest
Create Member	A	NA	NA
Terminate Other Member	A	NA	NA

Index

Account	pgs. 2, 3, 5, 7, and 9
Admin	pgs. 2, 4, 6, 7, 8, 11 and 12
Advanced search	pg. 4
Apportioning of requirements	pgs. 5 and 6
Assumptions and dependencies	pg. 5
External interfaces	pg. 6
Glossary	pg. 2
Guest	pgs. 2, 4, 6, 11 and 12
Introduction	pg. 1
Limitations	pg. 5
Member	pgs. 2, 3, 4, 6, 7, 9, 11 and 12
Performance requirements	pg. 11
Permissions	pg. 3, 4, and 5
Product perspective	pg. 3
Project repository	pg. 3
Purpose	pg.1
Scope	pg. 1
Specific requirements	pg. 6
TE³D house	pgs. 1, 2, and 3
Usability requirements	pg.10
User characteristics	pg.4