**CS673 Software Engineering** 

**Team 1 - Terriers Shopping Mall Product**

**Project Proposal and Planning**

| Team Member | Role(s) | Signature | Date |
| --- | --- | --- | --- |
| Haoran Dai | Team & Security | *Haoran Dai* | Oct.19. 2023 |
| Xiaocheng Kang | Requirement & Configuration | *Xiaocheng Kang* | Oct.19. 2023 |
| Qinchen Gu | Q & A | *Qinchen Gu* | Oct.19. 2023 |
| Bhargav Sai | Implementation | *Bhargav Sai* | Oct.19. 2023 |
| Huanzhou Wang | Design | *Huanzhou Wang* | Oct.19. 2023 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Revision history**

| **Version** | **Author** | **Date** | **Change** |
| --- | --- | --- | --- |
| **1** | **Xiaocheng Kang** | **Sept.27.2023** | **OCT.6.2023 (Updated)** |
| **2** | **Xiaocheng Kang, Qinchen Gu, Haoran Dai, Huanzhou Wang** | **Oct.18.2023** | **Oct.19 2023** |
| **3** | **Xiaocheng Kang, Qinchen Gu, Haoran Dai, Huanzhou Wang, Sai** | **Oct.26.2023** | **Nov.9.2023** |
|  |  |  |  |

[Overview](#_heading=h.gjdgxs)

[Related Work](#_heading=h.30j0zll)

[Proposed High level Requirements](#_heading=h.1fob9te)

[Management Plan](#_heading=h.3znysh7)

[Objectives and Priorities](#_heading=h.2et92p0)

[Risk Management (need to be updated constantly)](#_heading=h.tyjcwt)

[Timeline (need to be updated at the end of each iteration)](#_heading=h.3dy6vkm)

[Configuration Management Plan](#_heading=h.1t3h5sf)

[Tools](#_heading=h.4d34og8)

[Deployment Plan if applicable](#_heading=h.2s8eyo1)

[Quality Assurance Plan](#_heading=h.17dp8vu)

[Metrics](#_heading=h.3rdcrjn)

[Code Review Process](#_heading=h.26in1rg)

[Testing](#_heading=h.lnxbz9)

[Defect Management](#_heading=h.35nkun2)

[References](#_heading=h.1ksv4uv)

[Glossary](#_heading=h.44sinio)

# Overview

(Please give an overview of your project. It should include the motivation, the purpose and the potential users of the proposed software system, the basic functionality of the proposed software system and the possible technology stack to be used. )

For our project, we want to build a website about marketing.

For example, a website looks like Amazon, but all products related to BU, and BU alumni can post or buy used . Our shopping mall will be called Terriers Shopping Mall Products. Our main products should be new products and the web crawler, which we will crawl the website such as Barnes & Noble website from Boston University (<https://bu.bncollege.com/>), and admin uploads, and the used products from every user, such as faculties, students, etc.

For now, we will use Java and MySQL to do our project.

After we complete the backend and frontend codes, we need to combine them and make sure our project can run successfully. Then it will become fully complete.

# Related Work

(Please describe any similar software systems that you have found through the online research, and the differences between your software and those software systems.)

SQL database might be a selection. We did some research on the Barnes & Noble website, and Amazon. We want to make a website look like them. Also, we found that most of them only sell new products on their website, in the other words, we cannot find any things that will be cheaper because they are all new. Our website will be different from them, for sure we will have new products to sell, and we wanna give a way which can help students, faculties to upload their unused products or some products which are only used for once or twice. We also have a way to offer a reward to students or faculties. This part will look like Twitter and Facebook. Someone will post a forum to let others know what he/she needs, like we talked in a presentation. How does it work? Someone forgets something, and posts a forum to let others help, and the poster might give some rewards for the one who helped him/her, like some fees. We are thinking these parts might become better than others.

# Proposed High level Requirements

* 1. Functional Requirements  
     (For each functional requirement, please give a feature title and a brief description using the following format: As (a role), I want to (action), so that (value).)
     1. Essential Features (the core features that you definitely need to finish):

(For each essential features, please give a rough estimation in terms of person hours or an range of person hours)

As a buyer, I want to register an account by using username or email address, so that I can have access to all the products which will be placed on the website. (5-10 hours)

As a buyer, I want to log into the website with a password, so that I can come back anytime when I need something and to see if there is a suitable product which I want. (5-10 hours)

As a buyer, when I find a product that I like, I often want to see what else the seller has to offer. Having an option to view all items from the same seller can save time and might lead to me finding more products that I'm interested in purchasing. (5-10 hours)

As a buyer, I want to streamline my shopping experience by filtering products based on specific criteria (e.g., price, condition, category, brand). This way, I can quickly find the items that meet my requirements without going through every listing. (10-15 hours)

As a seller, I want to post some furnitures, books which I do not need anymore, so that I can log into websites and sell my furniture and books at a lower price to see if anyone else needs it. (5-10 hours)

As a buyer, I want to buy furniture and books, I added them into my cart to let them store together, so that I can go to cart and click checkout button, make a payment (Paypal, Credit Card, etc) to let managers know I bought that

As a customer, I wanna search for a specific product like a pillow, I can type it into the search bar, then it will show all the products which are pillows, so that I can find the one which I love, add to cart and checkout. (10-20 hours)

As a manager, I wanna let everyone have permission to search, it doesn’t matter if the user is logged in or not. So that everyone can see what we have and consider whether to sign in or not.

As a manager, I wanna withdraw the sales which are stored in the website. I can add my own payment into the website, so that I can receive sales from the website into my own payment account. (5-10 hours)

As a seller, I want to easily list multiple items at once, including bulk uploads of photos and descriptions, so that I can save time and effort in creating listings for all my products. (5-10 hours)

As a seller, I want to have a dashboard that shows my active listings, pending sales, and sales history, so that I can easily manage and track all my selling activities in one place. (5-10 hours)

* + 1. Desirable Features (the nice features that you really want to have too):

As a student with back-to-back classes, I want to post a reward for someone to help me print my important materials, so that I can have the necessary materials on time without missing any of my classes. (10-20 hours)

As a freshman in BU, it's really hard to buy furniture. Those days are moving dates for nearly everyone in Boston, so it is hard to buy used furniture, new ones are too expensive. I want to set up a website, so that people especially students can easily buy used cheap but still awesome furnitures. (10-20 hours)

As a buyer looking for textbooks, I want to search for books by ISBN or course code, so that I can find the exact textbooks I need for my classes more efficiently. (5-10 hours)

As a busy user, I want to set up alerts for specific items I'm looking for, so that I'm notified when they become available and don't have to constantly check the site. (5-10 hours)

* + 1. Optional Features (additional cool features that you want to have if there is time):
    2. Existing Features (delete this item if your project starts from scratch)
  1. Nonfunctional Requirements
     1. Security requirements

# Management Plan

## Objectives and Priorities

(Please describe your project objectives with highest priority first. Project Goals can include but not limited to complete all proposed (essential) features, deploy the software successfully, the software has no known bugs, maintain high quality, etc )

The highest priority for our project at this time should be database things, like for a website, the most important part is to find a database to support the website, as we first discussed together, we wanna use MongoDB to support our website. Then after discussion with the professor and TA, MongoDB might not be very helpful for securing, thus we are now trying to use a MySQL database to support our website. Then the template web also should be complete. For now, we are trying to complete a website with sign-in and sign up. In the future, we will add more to let it become a great website.(Will edit this part when we are reaching there) After completion, testing, debugging should be the important one. To make sure our website is going well.

## Risk Management (need to be updated constantly)

(Please write a summary paragraph about the main risks your group identified and how you plan to manage these risks. Then use the separate google sheet for detailed risk management. The template is provided in the same folder with this file. Please provide the link to the sheet.)

We have one member out, so we need to divide our parts again. And for the security problems, we need to change our database. Also we have some problems having meetings with all members, that makes some information not be very clear for that member.

First time to upload the Risk Management Sheet, will add the link before iteration 1.

**Risk Management Sheet Link:**

## Timeline (this section should be filled in iteration 0 and updated at the end of each later iteration)

| Iteration | Functional Requirements(Essential/Disable/Option) | Tasks (Cross requirements tasks) | Estimated/real person hours |
| --- | --- | --- | --- |
| 1 | Website with sign in/sign up parts/ Complete MySQL Database | Complete the database things and figure out the template web | 45 |
| 2 | Shopping Forum / Checking | Checking that shopping forum is working, need to connect between backend and frontend | 60 |
| 3 | Whole shopping forum / Connected with backend and frontend | Need to complete the whole shopping forum. Make sure it is working when connecting backend and frontend | 60 |

# Configuration Management Plan

## Tools

(In this project, we will use Git and Github as the version control tools. Please also specify any other tools to be used, e.g. IDE tools, CI/CD tools, container tools, SAST or DAST tools, and any other DevOps tools)

Code Commitment Criteria

* Atomic Commits: Each commit should represent a single logical change or feature.
* Descriptive Messages: Use clear and concise commit messages that explain the purpose of the change.
* Coding Standards: Follow coding conventions and maintain consistent coding style.
* Testing: Ensure code is adequately tested, and all tests pass before committing.
* No Debug Code: Remove debug statements or unused code before committing.
* Code Review: All commits must pass a code review by peers.

Version Control:

* Git and GitHub for version control and collaboration

IDE:

* Use IntelliJ IDEA for Java and Spring Boot development

CI/CD:

* GitHub Actions for automating builds, tests, and deployments.

Database Management:

* MySQL Workbench and Datagrip for database design and management.

DevOps Collaboration Tools:

* Discord for team communication and collaboration.

Dependency Management:

* Use Maven for managing project dependencies in the application.
  1. Code Commit Guideline and Git Branching Strategy  
     (Please briefly describe criteria for the code commitment and the branching strategy used, e.g. what are the branches to be used, how the pull request will be used etc. Here is an article to give you some basic knowledge about different git branching strategies: <https://www.flagship.io/git-branching-strategies/>

**Branching Strategy**

We follow the Feature Branching Strategy:

Main Branch (or Master):

* Production-ready code.
* Direct commits to main are limited and must be approved by designated stakeholders.

Feature Branches:

* Each feature or task has its own branch created from main.
* Branch naming convention: feature/<feature-name> or task/<task-name>.
* Frequent commits and collaboration during development.

**Pull Requests**

* PRs require review and approval by at least one team member.
* A PR can only be merged if it has no conflicts with the target branch.

## Deployment Plan if applicable

(If you plan to deploy your application (e.g. your web application), briefly describe how you plan to deploy your application).

**Database Configuration:**

Ensure that the MySQL database is properly configured and accessible from the deployment environment. Set up any necessary schemas and initial data.

# Quality Assurance Plan

## Metrics

(Describe the metrics to be used in the project to measure the quality of your software. Each metric should be measurable and quantifiable. Examples of metrics include product complexity (LOC, # of files, # of classes, # methods, cyclomatic complexity, etc.) , defect rate (# of defect per KLOC), # of test cases, test case pass rate, cost (# of person hours used), # of user stories completed, etc. **The result of these metrics should be reported in the progress report/ iteration summary sheet.**)

| Metric Name | Description |
| --- | --- |
| cost | Cost of person hours used, personal |
| total cost | Cost of person hours used, for whole group |
| Number of files | Number of files in the whole project, can show product complexity |
| Number of test cases | Number of test cases prepared for the product |
| Test case pass rate | Rate of test cases passed, can show the error rate of product |

* 1. Coding Standard

(Describe any coding standard to be used)

1. Standard headers for different modules: Provide name of modules, date of creation, author of module in headers to help everyone know who created module and what is the module about.

2. Standard name for each file and module: To help everyone easy to find the part of module if needed, for example, file productManagement

3. Indentation: Proper indentation to make code beautiful, easy to read and easy to find.

4. Comment: The code should include comments for at least important lines, to help code easy to read, understand and modify.

5. Length of functions should be as short as possible

6. Properly manage files: To make code in each file not too long, and also make the number of files in the project in a suitable range.

## Code Review Process

(Everyone should review all documents to be submitted. Here you will mainly describe how the code review will be done. Who will review the code, e.g. design or implementation leader will review all code or team members review each other’s code. Do you use pull requests for the code review? Is there a checklist to help review? What feedback should the reviewer provide?)

1. Everyone should double check his own code, try to make sure everything is good when gathering.

2. Implementation leader will do the second check for all codes gathered.

3. Others should also check codes if time is available.

4. There should be a checklist for everyone to help review, the list can be determined in the later meeting by everyone.

5. Everyone who did the review should provide feedback to show his own opinion of the code, for example: The code is too redundant, has too much repetition, and that function can be packaged to be used to make code clear and easy to read and understand.

## Testing

(Both manual testing and automated testing should be considered. Both unit testing and integration testing should be considered. Briefly describe the testing tools/framework to be used, the personnel involved (e.g. the QA leader will focus on the integration testing and each developer will unit test their own code), when and what types of testing will be performed, the testing objectives, etc)

1. Everyone should test their own code before gathering, testing methods including but not limited to JUnit, QUnit, Jest. Also everyone’s own mind and hand will be another important part of test, for example for front-end, members can test by open pages in browser and check error codes in browser console, or check error code provided by IDE.

2. Q&A leader will focus on integration testing, to make sure every part of code also works well after gathering.

3. Tests should include as many test cases as possible, including but not limited to all buttons connected to the right function, all functions work well, all databases work properly.

## Defect Management

(Describe the tool to be used to manage the defect (e.g github issues). The types of defects to look at. The actions or personnel for defect management. )

1. Planning to use github issues as a defect management tool.

2. Everyone should try to find bugs in the whole project, and the Q & A leader should be the person who manages it.

3. Once defects are found, people who create this part of code will be directly responsible for the solution, and the implementation leader should help everyone who had a defect to find solutions.

# References

(For more details, please refer to the encounter example in the book or the software version of the documents posted on blackboard. )

[Coding Standards and Guidelines - GeeksforGeeks](https://www.geeksforgeeks.org/coding-standards-and-guidelines/)

<https://git-scm.com/doc>

<https://zhuanlan.zhihu.com/p/79413784>

<https://docs.github.com/en/actions>

<https://cloud.spring.io/spring-cloud-config/reference/html/>

# Glossary

(Any acronym used in the document should be explained here)